ADAPTIVE CONTROL FOR MODULAR VERTICAL LIFT AIR VEHICLES

A Thesis
Presented to
The Academic Faculty

by

Brady Anderson

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2017

COPYRIGHT © 2017 BY BRADY ANDERSON

ADAPTIVE CONTROL FOR MODULAR VERTICAL LIFT AIR VEHICLES

Approved by:

Dr. Jonathan Rogers, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Aldo Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Jun Ueda
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: November 14, 2017

To the students of the Georgia Institute of Technology

ACKNOWLEDGEMENTS

I thank my wife and son for their continued dedication and encouragement in helping me finish this work. I would like to thank my mother and father, for their support. I also thank my brother for introducing me to the awesome world that robotics is. Finally, I am grateful for Dr. Rogers and the iREAL lab at Georgia Tech for aiding and enhancing my understanding and ability to create this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

α_{init}	initial nominal angle of attack provided by user
$\Delta\alpha$	correctional angle of attack to optimize lift
A	planform area
A_s	sideforce area
β_i	pitch angle assigned to *ith* gimbal
A_{Vi}	diagonal matrix of the frontal areas for each axis
$\left(A_{x,Vi}, A_{y,Vi}, A_{z,Vi}\right)$	components of A_{Vi}
c_T	thrust coefficient
c_Q	torque coefficient
C_l	lift coefficient
$C_{D,Vi}$	diagonal matrix of vehicle drag coefficients for the vehicle
$\left(C_{Dx,Vi}, C_{Dy,Vi}, C_{Dz,Vi}\right)$	components of $C_{D,Vi}$
$C_{d,P}$	drag coefficient for the payload, (or airfoil)
C_s	the (sideforce) coefficient of drag
C_L/C_D	ratio of lift to drag
CG	center of gravity
e_{Lift}	error in lift magnitude
$\vec{e}_{\Omega}$	orientation derivative error
$\vec{e}_{ori}$	error in orientation
$\vec{e}_{xyz}$	error in inertial frame position
e_{Th}	position error threshold beyond which controller becomes velocity tracking only

$\vec{f}$	augmented state dynamic equations
F	Jacobian of augmented state dynamic equations
$\vec{F}_{Drag,Vi}$	force of drag on *ith* vehicle
$\vec{F}_{Lift}$	force of lift on *ith* vehicle
$\vec{F}_{Drag}$	drag on payload or airfoil
$\vec{F}_{L,D,S}$	total lift, drag, and side-force vector on airfoil
$\vec{F}_{P}$	total force on payload
$\vec{F}_{Sideforce}$	side-force vector on airfoil
$\vec{F}_{Vi}$	total exerted force of vehicle on payload
g	gravity
G	process noise influence matrix
γ_i	roll angle assigned to *ith* gimbal
Γ_W	angle of dihedral for wing frame
$\vec{h}$	measurement equations
H	Jacobian of the measurement equations
i_W	angle of incidence for wing frame
I	system inertia matrix
$\vec{I}_F$-$\vec{J}_F$-$\vec{K}_F$	component unit directions to define frame of subscript F
$k_{p,\alpha}$	proportional gain coefficient for angle of attack control
$k_{d,\alpha}$	derivative gain coefficient for angle of attack control
K	Kalman gain
K_p	diagonal proportional gain matrix for outer loop
$K_{p,in}$	diagonal proportional gain matrix for inner loop

K_d	diagonal derivative gain matrix for outer loop
$K_{d,in}$	diagonal derivative gain matrix for inner loop
L_F	a projection onto the $\vec{I}_F$-$\vec{K}_F$ plane of frame F
Λ_W	angle of sweep for wing frame
M	mass of composite vehicle (payload plus attached vehicles)
$mag_{r,Vi}$	magnitude of the x-y component of radius vector
$\vec{M}_{Des}$	output inner loop control law, desired moments on body
L_{Des}, M_{Des}, N_{Des}	components of $\vec{M}_{Des}$
$\vec{M}_{dir,Vi,pre}$	resultant unit torque direction from 1st step
L_{Vi}, M_{Vi}, N_{Vi}	components of $\vec{M}_{dir,Vi,pre}$
$\vec{M}_{Vi}$	moment on vehicle
$\vec{M}_P$	moment on payload
n	number of vehicles attached to the payload
$\vec{N}_{dir,Des}$	unit vector in direction of corrective thrust for step two
N_{res}	residual yaw moment magnitude to be resolved in step two
$\vec{N}_{res}$	vector of residual yaw to be resolved in step two
ω_j	rotor angular velocity
$\vec{p}$	vector of parameters to be estimated
P	state error covariance matrix
P_F	projection onto 2D space, specifically onto the *I-J* plane of frame F
q_0	scalar portion of quaternion vector
q_1, q_2, q_3	vector portion of quaternion vector
Q	process noise covariance matrix

ρ	air density
R	measurement noise covariance matrix
$r_{x,y,nom}$	nominal radius *x*-*y* location on payload, randomized
$\mathrm{r}_{x,y,\oplus P\to Vi,Dist}$	attach point distribution
$\vec{r}_{\oplus Vi\to Vi,dj}$	radius from vehicle CG to *jth* rotor disk
$\vec{r}_{\oplus P\to Vi}$	Radius from payload CG to *ith* vehicle
(x, y, z)	inertial frame position vector components
$(x, y, z)_{\mathrm{Des}}$	desired inertial frame position
(ϕ, θ, ψ)	body fixed rotation angles, 3-2-1 method
(u, v, w)	body fixed velocity vector components
(p, q, r)	body fixed angular velocity vector components
(X,Y,Z)	total force on body components
(L,M,N)	total moment on body components
$R_{F0\to F1}$	rotation matrix from frame F0 to frame F1
S_F	projection onto the y-axis of frame F
$\vec{\tau}_{dj}$	yaw torque from one rotor on vehicle
$\vec{T}_{dj}$	thrust from jth rotor disk on a vehicle
$\vec{T}_i$	thrust from *ith* vehicle
$\vec{T}_{Des}$	desired thrust vector output created by outer loop control law
T_{mag}	magnitude of thrust vector to be resolved by kinematic inversion
$T_{Vi,pre}$	thrust magnitude assigned to *ith* vehicle in first step
ΔT_{Vi}	corrective thrust magnitude for *ith* vehicle in step 2
$\vec{T}_{Vi,final}$	final thrust vector for *ith* vehicle
v_{max}	user define maximum velocity to allow during flight

$\vec{V}$	velocity vector of system
$\overrightarrow{W}$	composite system weight
$\vec{x}$	full state vector
$\vec{x}_A$	augmented state vector
$\Delta\vec{x}_{A,est}$	estimate correction state vector
$\vec{y}_m$	measured state output
σ^2_{mult}	standard deviation error multiplier
μ_{mult}	mean error multiplier
η_{mult}	mixed error multiplier
	Subscripts
I	inertial frame
P	payload frame
V	vehicle frame
W	wing frame
Vi	*ith* vertical lift vehicle
L	lift
D	drag
des	desired
std	standard deviation error distribution
mean	mean error distribution
mix	mixed error distribution
est	estimated
i	*ith* (vehicle)
j	*jth* (individual rotor)

SUMMARY

Modular vertical lift air vehicles are comprised of individual vertical lift vehicles connected to a payload for the purposes of cooperative manipulation and transportation. This method of payload manipulation has unique advantages in terms of scalability, flexibility, and robustness to vehicle failure due to redundancy. However, there are numerous difficulties with designing and controlling such systems ranging from mechanical complexity of attachment devices to control allocation and robustness to uncertainty. This thesis explores various aspects of modular air vehicle design and control. A simulation model is first described which supports high-fidelity simulation of vehicle-payload attachment, payload transportation, and vehicle-payload detachment. A control allocation scheme is also proposed for cooperative payload manipulation. Simulation studies using this control allocation scheme examine flight performance and energy consumption in example scenarios where modular vertical lift vehicles cooperatively transport various payloads, including bluff bodies and airfoil shapes. Further studies analyze effectiveness of an extended Kalman filter in providing key geometric and mass center location values to the control scheme. Finally, the mechanical design of a modular vertical lift vehicle is provided, including experimentally-derived estimates of thrust and energy consumption.

CHAPTER 1. INTRODUCTION

Much work has been done in recent years studying aspects of modeling, simulation, and control of quadrotor vehicles. The motivations for this work range widely from photography to crop-dusting and surveillance for law enforcement. Quadrotors offer unique advantages in terms of mechanical simplicity, ease of anti-torque application, vertical flight capability, and relatively compact design [1-3]. However, to date, the vast majority of research in the quadrotor domain has targeted applications in which the vehicle carries a very light payload – for instance, cameras or other small sensors. This is primarily due to poor scaling of thrust and power relationships with rotor size. Hover efficiency, measured in terms of thrust per unit power, grows hyperbolically with rotor disk area [4], meaning that smaller vehicles are practically limited to carrying only very light loads (given current battery or miniaturized engine technology). These loads are much lighter than those carried by large single or tandem rotor helicopters, and thus standard helicopters have maintained their dominant role as the vehicle of choice for many cargo transportation missions.

Nevertheless, it is possible that multiple vertical lift vehicles can collaborate to lift payloads on the scale of those currently transported by larger aircraft. Potentially, multiple vertical lift vehicles may fly independently, converge on a payload, and attach to it, forming a so-called composite vehicle. The composite vehicle then flies to a destination under the collaborative control of the individual attached vehicles and lands. After landing, the vehicles can detach and perform another mission. By using modular vehicles in this fashion, a heavy-lift capacity may be highly portable and distributed, allowing for

redundancy in cases of vehicle failure. In some cases, payloads may even be specifically designed with an airfoil shape to reduce thrust and energy requirements in forward flight. While potentially transformative, numerous technical issues exist with this idea including problems of scalability, energy consumption, cooperative control, modeling and simulation, and attachment mechanical complexity, among others. This thesis presents a study of the overall topic of cooperative transport via modular vertical lift, exploring issues of modeling, simulation, control, and preliminary mechanical design.

From a modeling perspective, numerous simulation tools of varying degrees of accuracy have been developed to describe the dynamic behavior of quadrotor vehicles (for example [2, 5-11]). However, all of these works consider the dynamics of a single vehicle only. There have also been several investigations into swarm systems [6, 8, 12], but these again treat the vehicle dynamics only in an individual fashion and would be applicable for the modular lift system only when the vehicles are in individual (disconnected) flight. Recent work described in [9] is the closest related work, studying cooperative control of quadrotor vehicles which are collectively grasping and lifting a payload. However, many assumptions are made that reduce the capabilities of the system. For instance, one of the main assumptions used in [9] is that the weight of the payload is much less than that of the vehicles. This allows the location of the payload center of gravity to be ignored. The present work considers loads approaching several times the mass of the vehicles, making it impossible to ignore the payload center of gravity and inertias. Other related work, such as in [5] and [10], use cables to suspend the load beneath multiple vehicles. However, cables can induce severe vibrational modes into the system that can make cooperative flight extremely difficult without precise knowledge of system parameters. In a practical system,

it is unlikely that such perfect knowledge would be available, and thus the current work considers rigid or hinged attachments only.

On the subject of control allocation, there have been many groups working on assigning redundant systems efficient control output commands, or to counteract actuator failures [13-17]. Others are focused on widening the capabilities of a typical quadrotor by introducing control authority in the body *x*-*y* plane [18-20]. The current work adds to these efforts by using a minimum norm inverse solution, and by allowing the thrusts to be directly vectored as desired. The controller here has greater impact in that in can be applied to larger systems of multiple vehicles carrying a payload, not just a single vehicle's rotors being vectored to carry out a light-weight task.

Using the inverse as described in this work is similar to the work in [15, 21-22]. Positive results of using the inverse to find the minimum actuation necessary are clearly demonstrated in Chapter 4. Further, the present work focused on lightly modifying an existing, well-known quadrotor controller to make it applicable to a scalable, modular system of any number of thrusters attached to a payload. The two-step approach to accomplish this with less computationally intensive inversions can be applied to any over actuated system powered by vectored thrusters.

This work describes the development of a simulation tool for modeling cooperative load manipulation and transportation using modular vertical lift vehicles. Both individual and composite vehicle models are described, including the important assumptions that underlie each model. Next, a control algorithm is described for stabilization and control of the composite vehicle. This controller is based on a set of tracking controllers combined

with a control allocation scheme. Additionally, an extended Kalman filter is described that allows the real-time estimation of important values used in the control allocation steps, removing the necessity of knowing payload parameters beforehand.

A variety of simulation results are presented to predict performance in several illustrative scenarios. One set of results demonstrates an example trajectory in which four vehicles converge on and attach to a rigid payload, and collaboratively transport it several hundred meters. To demonstrate possible use of this system in casualty evacuation, this payload is modeled on the size and weight characteristics of a typical human. Furthermore, a series of simulation studies analyze the stabilization characteristics of the flight control scheme when presented with imperfect knowledge of payload parameters. This analysis is performed both with and without the Kalman filter real-time parameter estimator.

Additional simulation results consider power consumption of the combined set of vehicles performing an example mission with payloads of varying aerodynamic characteristics. Specifically, performance of the system using a bluff body payload versus a payload shaped as an airfoil are compared. Results show significant performance benefits can be obtained when using airfoil-shaped payloads, especially if vehicle attachments permit sufficient flexibility to fly the payload at its most efficient angle of attack.

Finally, to demonstrate the feasibility of such a system, the mechanical design of an 8-rotor modular vertical lift vehicle is presented. This vehicle is shown to have a total thrust capacity of 90 pounds and a payload capacity of 65 pounds. Experimentally-obtained thrust and power estimates show that 3-4 of these vehicles would have sufficient lift capacity and

endurance to be able to perform a casualty evacuation mission similar to the one shown in simulation results.

CHAPTER 2. VEHICLE SIMULATION MODEL

The system model employed in this paper operates in two modes. In individual vehicle flight mode, the model simulates the flight of n independent multirotor vehicles. In composite vehicle mode, the n vehicles are assumed to be rigidly attached to the payload via gimbal-actuated joints. While such a system can be modeled using various multi-body simulation techniques, in this work it is assumed that vehicle rotational motion with respect to the payload is quasi-static only. Thus, the entire composite vehicle consisting of the payload and attached vehicles can be modeled as a single 6-degree-of-freedom (6DOF) rigid body. Throughout this work, the standard north-east-down inertial reference frame is used with the associated flat-earth approximation for all modeling and simulation developments.

There are numerous possible ways to simulate vehicle attachment and detachment to and from the payload. In this paper, vehicle attachment to the payload is assumed to be initiated once the vehicle has achieved a stable hover within a small threshold distance to the specified payload attachment point. Once this criterion is satisfied, the independent vehicle model is deactivated, and its inertial characteristics are combined into the composite vehicle to signify rigid attachment. For all results in this paper, all individual vehicles must be attached to the composite vehicle before composite vehicle flight is initiated. Vehicles detach from the payload in an instantaneous fashion, during which the independent vehicle model is reactivated using the last state of the composite vehicle (with kinematic corrections for the vehicle attachment point). The following subsections provide an in-depth discussion of both of the individual and composite simulation modes, including all relevant equations.

Before giving the in-depth modeling formulations for the individual and composite vehicles, descriptions of pertinent frames and vectors used are related in Figure 1-Figure 4. Figure 1 shows the principle reference frames used for this simulation, and as shown, the north-east-down scheme is used throughout the simulation for the inertial frame. An inertial frame is denoted with subscript I, $\vec{I}_I$-$\vec{J}_I$-$\vec{K}_I$, and payload and vehicle frames are similarly denoted with subscripts P and V respectively. Figure 2 shows an optional wing frame, denoted by subscript W, that can be defined separately from the payload frame, in the case that the wing is defined having some incidence, i_W, sweep, Λ_W, and dihedral angles, Γ_W. Having this extra frame aids in calculation of the aerodynamic effects of the wing.

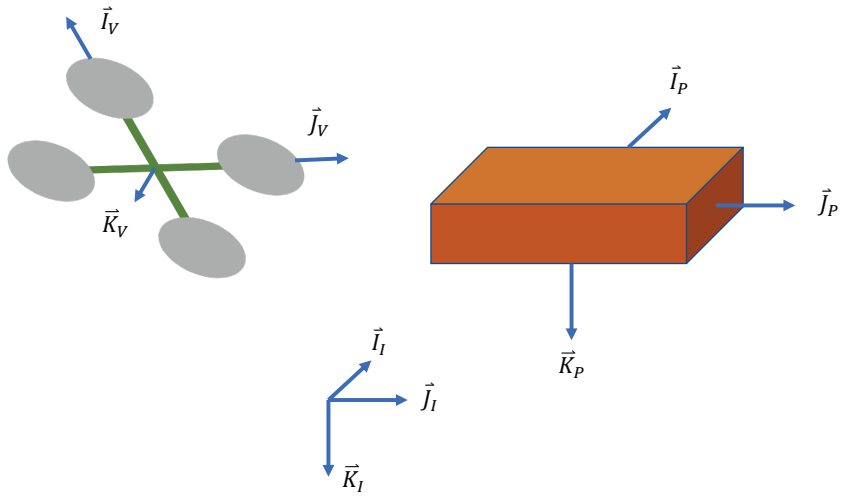

Figure 1 – Reference Frames. Inertial, payload, and vehicle frames are shown with respective relations to each other

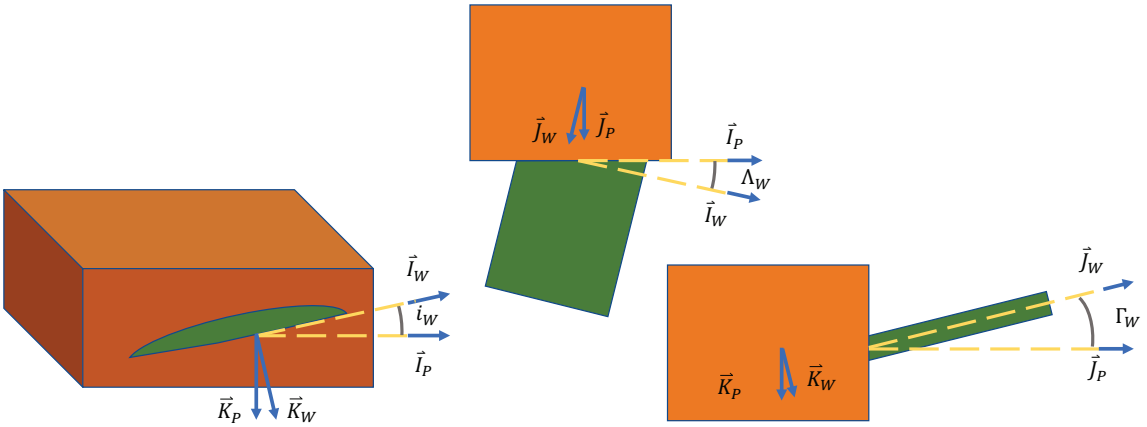

Figure 2 – Wing Frame. Optional wing frame offset from payload frame for aerodynamic calculations.

Figure 3 is a schematic of a typical quadrotor, with notation of rotor and thrust directions and their numbering scheme. Following, Figure 4 shows the so-called 'radius-vectors' from the payload CG to the vehicle CG. The vehicles are connected to the payload by means of an arm with two links. The lower arm is rigidly attached to the payload, and the upper arm is rigidly attached to the vehicle. The joint between them has an actuator capable of inducing some pitch and roll to the vehicle frame from the payload frame. It is assumed throughout the simulation that the upper link is short enough that it does not significantly alter the radius vector to the payload CG after any non-zero pitch and roll commanded to the joint.

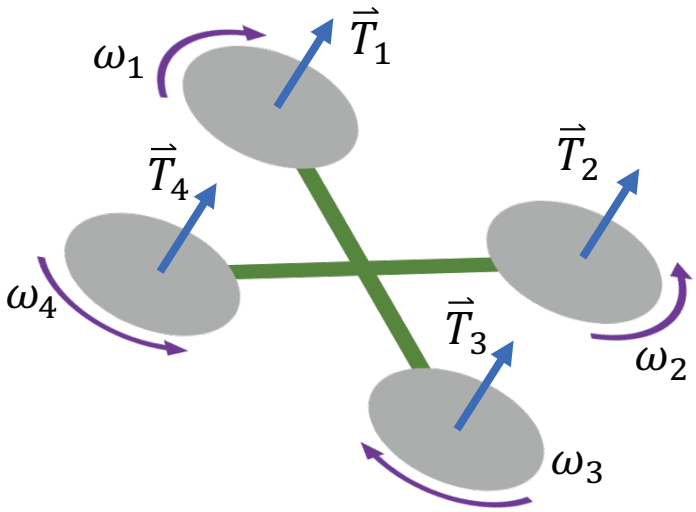

Figure 3 – Quadrotor Vehicle Schematic. Rotors are numbered as shown. Thrusts, $\vec{T}_j$, and rotor angular velocities, ω_j, are noted. The first rotor lies along $\vec{I}_V$, and the second rotor lies along $\vec{J}_V$.

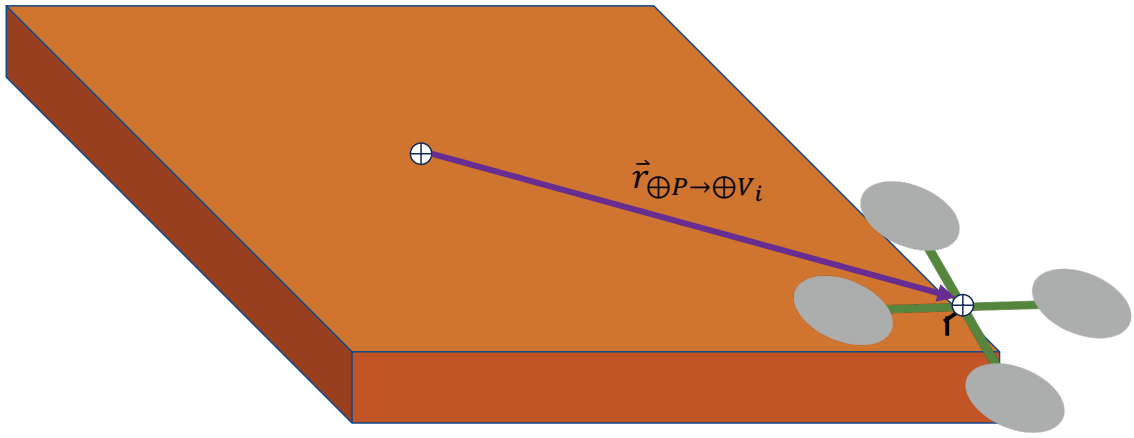

Figure 4 – Radius Vector. The vector defined from the payload CG to each individual vehicle is a radius vector.

2.1 Individual Vehicle Modeling

When each vehicle is in individual flight (i.e., not connected to the payload), a standard multirotor model may be used to simulate motion. The model used in the simulation described here is based on the 6DOF model developed in [3]. The twelve states used to describe the vehicle's motion are (x, y, z) representing the vehicle position in the inertial frame, (ϕ, θ, ψ) representing the vehicle Euler angles, (u, v, w) representing the body-frame translational velocity components of the mass center, and (p, q, r) representing the body-fixed angular velocity components. The relevant dynamic equations for this model, as well as expressions for the forces and moments acting on the body, can be found in [23]. Additional details regarding the 6DOF multirotor vehicle model employed here are omitted for brevity and the reader is referred to [3] and [23].

The only major deviation of the model used in this work compared to the models provided in [3] and [23] is in calculation of the drag forces on the multirotor. The present work utilizes a bluff body drag model as show in Equation 1, approximating drag coefficients for 3D-prisms from those provided in [24].

$$\vec{F}_{Drag,Vi} = -0.5\,\rho\,\left\|\vec{V}_{Vi}\right\|\,C_{D,Vi}\,A_{Vi}\,\vec{V}_{Vi} \tag{1}$$

In Equation 1, ρ is the air density, the subscript Vi refers to the vertical lift vehicle, and $\vec{V}_{Vi}$ is the vehicle velocity vector, $[u, v, w]^T$. Because the frontal area is larger in the $\vec{K}_V$ direction, the coefficients $C_{Dz,Vi}$ and $C_{Dxy,Vi}$ differ. Therefore, $C_{D,Vi}$ is the diagonal matrix of these vehicle drag coefficients for the vehicle, and similarly, the matrix A_{Vi} is the diagonal matrix of the frontal areas for each axis, summarized in Equations 2-3.

$$C_{D,Vi} = \text{diag}\left(C_{Dx,Vi}, C_{Dy,Vi}, C_{Dz,Vi}\right) \tag{2}$$

$$A_{Vi} = \text{diag}\left(A_{x,Vi}, A_{y,Vi}, A_{z,Vi}\right) \tag{3}$$

The simplification in the drag model, compared to that used in [3], was made due to the unique nature in which the vehicles are to be used while attached to the payload. While flying unattached, the drag model described in Equation 1 and that described in [3] only vary by about fifteen percent. While attached to a payload in forward flight, the vehicles are flying in an orientation such that most of the velocity is directed along the $\vec{K}_V$ axis. This near-axial flight condition is very different from that considered during model development in [3]. The relatively small deviations of computed drag in solo flight, combined with the unconventional use of the vehicles in an axial fashion when attached to an airfoil payload, give reason to use this more generalized drag model throughout the simulation cases presented here.

Whereas the model described in [3] and [23] applies generally to any multirotor vehicle, some modeling and control aspects in this work are specific to a unique type of octocopter (8-rotor vehicle) arranged with two rotors in a coaxial configuration on four arms. This design is effectively a quadrotor, and is employed to achieve maximum thrust capability with minimal increase in vehicle size. Although the rotors on each arm are counter-rotating, differences in inflow between the upper and lower propellers means that the torque from each coaxial rotor pair will not always balance to zero. Because each rotor has a specific relationship between the thrust coefficient and the (nonzero) torque coefficient, a quad-rotor model can be used to represent this vehicle without loss of generality or fidelity. Additionally, whereas a traditional quadrotor controller uses a torque differential between two pairs of counter-rotating rotors, in practice the co-axial

configuration allows for almost double the yaw torque capability of a typical quadrotor. This is due to the presence of four pairs of counter-rotating rotors. As discussed later, these thrust and torque parameters of the vehicle considered here are determined experimentally using a prototype vehicle implementation.

2.2 Composite Vehicle Modeling

The composite vehicle is modeled as a 6-degree-of-freedom rigid body. The external forces applied to this body include the combined thrust of all *n* attached vehicles, the combined weight of the vehicles and payload, the aerodynamic forces acting on the attached vehicles, and the aerodynamic forces acting on the payload. Likewise, the total moment about the mass center is comprised of contributions from all rotor thrust values (given their relative position with respect to the composite payload center of gravity), and aerodynamic moments on the payload itself. For the purpose of this model, the individual commanded vehicle thrust vectors are assumed to act at the vehicle-payload attachment points, which are also assumed to be at the centers of the respective multirotor vehicles. The same aerodynamic model used to compute drag on the individual vehicles in free flight is employed to compute vehicle drag when connected to the payload. Quaternions are used as the rotation parameters describing the orientation of the payload and vehicles with respect to the inertial frame. In this work, q_0 describes the scalar portion of the quaternion and q_1, q_2, and q_3 to describe the directional portion of the quaternion vector.

Equation 4 describes how the weight is resolved in the payload body frame, where $\overrightarrow{W}$ denotes the composite system weight, *m* is the mass of the composite vehicle, including the mass of all attached vehicles and the payload, and *g* is used to denote gravity.

$$\vec{W} = m\,g \begin{bmatrix} 2\,(q_1 q_3 - q_0 q_2) \\ 2\,(q_2 q_3 + q_0 q_1) \\ {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix} \tag{4}$$

In addition to the individual vehicle drag, aerodynamic forces are also exerted on the payload. The payload aerodynamic model used here is highly general, allowing for the computation of drag on a bluff body payload, or optionally computing drag and lift forces as a function of payload angle of attack. The latter allows the payload to be modeled as a variety of shapes, particularly including airfoils. Equations 6 and 7 relate the lift and drag force vectors acting on the composite vehicle. The model allows for an angle of incidence, and sweep, both set by the user upon initialization. A transformation matrix, $R_{P\to W}$, is used to transform vector components from the payload frame into the wing frame for force calculation. Here, C_l is the lift coefficient, A is the planform area of the payload's wing (or wings), $C_{d,P}$ is the drag coefficient, and $\vec{V}_P$ is the payload velocity vector.

Equation 5 defines a matrix L_F that is a projection onto the $\vec{I}_F$-$\vec{K}_F$ plane of a frame F. This is used to compute the total lift and drag force vectors, $\vec{F}_{Lift}$ and $\vec{F}_{Drag}$, in Equations 6-7.

$$L_F = \left[\, \vec{I}_F \quad \vec{0} \quad \vec{K}_F \,\right] \tag{5}$$

$$\vec{F}_{Lift} = -0.5\,\rho\,C_l\,A\,\left\|\vec{V}_P\right\|\left(L_W\,\vec{V}_P \times \vec{J}_W\right) \tag{6}$$

$$\vec{F}_{Drag} = -0.5\,\rho\,C_{d,P}\,A\,\left\|\vec{V}_P\right\|\,L_W\,\vec{V}_P \tag{7}$$

Under the assumption of no wind, the side-force acting on the wing is computed Equation 9 with a general bluff body approach, similar to Equation 1. The notation C_s is used to denote the coefficient of drag along $\vec{J}_W$, and likewise A_s is the frontal area facing the same direction. To facilitate the calculation, a projection onto the $\vec{J}_F$ direction of a frame, F, a matrix S_F is defined by,

$$S_F = \begin{bmatrix} \vec{0} & \vec{J}_F & \vec{0} \end{bmatrix} \tag{8}$$

$$\vec{F}_{Sideforce} = -0.5\,\rho\,C_s\,A_s\,\left\|\vec{V}_P\right\|\,S_W\,\vec{V}_P \tag{9}$$

Once the lift, drag, and side-force vectors have been computed in Equations 6, 7, and 9, they are summed in Equation 10 for a total aerodynamic force vector acting on the composite vehicle,

$$\vec{F}_{L,D,S} = \vec{F}_{Lift} + \vec{F}_{Drag} + \vec{F}_{Sideforce} \tag{10}$$

In order to calculate the coefficients of lift and drag for the airfoil, a lookup table is used with the current angle of attack, α, calculated from the velocity vector. The values used in this work are for a NACA0012 airfoil. Once a value for the angle of attack is found, using Equation 11, the appropriate C_l and $C_{d,p}$ are linearly interpolated.

$$\alpha = \tan^{-1}\left(\left(\vec{V}_P \cdot \vec{K}_W\right)/\left(\vec{V}_P \cdot \vec{I}_W\right)\right) \tag{11}$$

To develop the forces and moments that each vehicle exerts on the composite vehicle, it is important to mention that quasi-static thrust vectoring is used for each vehicle

about its attachment point. A vehicle frame *Vi* is defined and obtained through a 2-axis, body-fixed transformation from the payload frame, and let $R_{P\to Vi}$ denote the transformation matrix from the payload frame to the *i-th* vehicle's body frame. The method of solving for the desired rotation angles of each vehicle will be discussed in the next section. The total force exerted on the payload due to each individual vehicle, $\vec{F}_{Vi}$, is given by Equation 13. It is the sum of the thrust vector from each of the vehicle's rotors as well as the drag vector from the vehicle aerodynamics. Each rotor's thrust, $\vec{T}_{dj}$, is given in Equation 12,

$$\vec{T}_{dj} = -c_T\ \omega_j{}^2\ \vec{K}_{Vi} \tag{12}$$

$$\vec{F}_{Vi} = \sum_{j=1}^{m} \left[\vec{T}_{dj}\right] + \vec{F}_{Drag,Vi} \tag{13}$$

where ω_j is the commanded angular velocity of the *j-th* motor on the vehicle and c_T is the thrust coefficient. Finally, adding all the forces together as shown in Equation 14, the total force on the payload is given by,

$$\vec{F}_P = \vec{W} + \vec{F}_{L,D,S} + \sum_{i=1}^{n} \left[\vec{F}_{Vi}\right] \tag{14}$$

The moments generated by each vehicle's rotors about its own mass center are given by L_{Vi}, M_{Vi}, and N_{Vi}, and are given by Equation 16. Each rotor's yaw torque $\vec{\tau}_{dj}$ also adds to the moment acting on the body, and is given in Equation 15,

$$\vec{\tau}_{dj} = (-1)^j\ c_Q\ \omega_j{}^2\ \vec{K}_{Vi} \tag{15}$$

$$\vec{M}_{Vi} = \sum_{j=1}^{m} \left[\vec{r}_{\oplus Vi \to Vi,dj} \times \vec{T}_{dj} + \vec{\tau}_{dj}\right] \tag{16}$$

where $\vec{r}_{\oplus Vi \to Vi,dj}$ is the radius of a rotor disk j from the vehicle's own mass center, and c_Q is the torque coefficient. Depending on the vehicle in use, the direction of spin of the first rotor will determine the power of (-1) to be used in Equation 15. If the first rotor has a positive spin, i.e. the angular velocity is positive, it will produce a negative torque on the vehicle. These moments sum with the moments created by the vectored thrusts, as shown in Equation 13, to complete the moments acting on the composite vehicle,

$$\vec{M}_P = \sum_{i=1}^{n} \left[\vec{M}_{Vi} + \vec{r}_{\oplus P \to Vi} \times \vec{F}_{Vi}\right] \tag{17}$$

where $\vec{r}_{\oplus P \to Vi}$ is the radius vector from the payload mass center to the vehicle mass center.

When simulating a trajectory of the composite vehicle, Equations 1-17 are used to compute the total forces and moments on the composite vehicle. The rigid body equations of motion of the composite vehicle are given by Equations 18-21, which are numerically integrated using a fourth order Runge-Kutta (RK4) method.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2\,(q_1 q_2 - q_0 q_3) & 2\,(q_1 q_3 + q_0 q_2) \\ 2\,(q_1 q_2 + q_0 q_3) & {q_0}^2 - {q_1}^2 + {q_2}^2 - {q_3}^2 & 2\,(q_2 q_3 - q_0 q_1) \\ 2\,(q_1 q_3 - q_0 q_2) & 2\,(q_2 q_3 + q_0 q_1) & {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ w \end{bmatrix} = \frac{1}{m} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = [I]^{-1} \left[\begin{bmatrix} L \\ M \\ N \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} [I] \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right] \tag{21}$$

In Equation 21, [*I*] is the inertia matrix of the composite vehicle about its mass center.

.

CHAPTER 3. CONTROL ALGORITHM

The control of this system is completed by using a proportional derivative (PD)-based feedback control law. As this controller relies on a kinematics inversion-based control allocation scheme, advanced knowledge is necessary in order to accurately produce desired thrusts and moments. Many instances may not allow for this type of knowledge about a payload to be transported. Therefore, an extended Kalman filter is added to the loop in order for certain critical parameters to be estimated. A description of the controller and filter follow. Figure 5 and Figure 6 show diagrams outlining the simulation and the controller's two-step process.

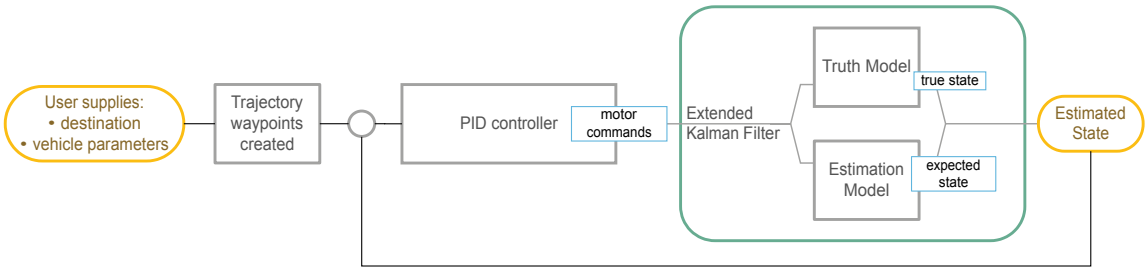

Figure 5 – Flow Chart of Simulation with Kalman Filter.

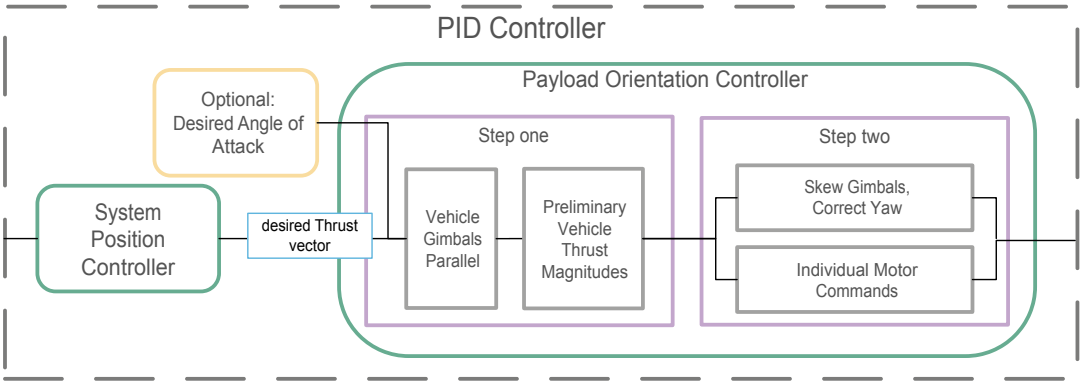

Figure 6 – Breakout of the PID Structure. The two-step process of the controller is detailed.

3.1 Individual and Composite Vehicle Control

In individual flight, the vehicles are controlled using the proportional derivative feedback control law described in [3]. This is a rather standard quadrotor control law available in several references, and thus will not be reiterated here. In composite vehicle flight, a two-step approach is used to compute the commanded thrust values for each attached vehicle based on a control allocation method.

To derive the composite control law, consider an inner-outer loop PD control structure in which an outer loop is used to compute a desired thrust vector $\vec{T}_{Des}$ given errors in position and velocity based on a desired waypoint provided by the user. The inner loop commands the vehicle orientation by means of a desired moment vector. This is a similar structure employed in the individual vehicle control laws of [3] and [23], and can be used with some modification for composite vehicle control. The user also provides a maximum velocity, v_{max}, that is not to be exceeded by the system, as well as an error threshold, e_{Th}, which is used to change the controller structure into a velocity-tracking mode. The positional error vector in Equation 23 is used to compute the magnitude of the error in position in the horizontal inertial plane, $\vec{I}_I$-$\vec{J}_I$, in Equation 24. First, a projection matrix onto the *I-J* plane of a denoted frame subscript F, P_F, is defined in Equation 22,

$$P_F = \begin{bmatrix} \vec{I}_F & \vec{J}_F \end{bmatrix}^T \tag{22}$$

$$\vec{e}_{xyz} = \begin{bmatrix} x - x_{Des} \\ y - y_{Des} \\ z - z_{Des} \end{bmatrix} \tag{23}$$

$$e_{xy} = \left\| P_I \ \vec{e}_{xyz} \right\| \tag{24}$$

The derivative of the positional error, $\dot{\vec{e}}_{xyz}$, is computed using the system velocity and desired velocity instead of positions as in Equation 23.

If the position error magnitude is greater than the provided e_{Th}, the proportional control is deactivated in Equation 26 for the inertial $\vec{I}_I$-$\vec{J}_I$ plane. The derivative control acts on both the weighted horizontal velocity error, described in Equation 25, and the vertical velocity error. Equation 25 shows the new commanded velocity vector used by the outer loop control law while in velocity tracking mode. Further, the altitude is the only position variable to be tracked when the system is in this mode. Therefore, the proportional gains for the horizontal desired trajectories are turned off by replacing the first two elements of the diagonal proportional gain matrix, K_p, with zero, while maintaining K_d unchanged.

$$\vec{V}_{Des,temp} = v_{max} \ e_{xy}^{-1} \ \dot{\vec{e}}_{xyz} \tag{25}$$

The projection of the temporary desired velocity vector, $\vec{V}_{Des,temp}$, onto the inertial $\vec{I}_I$-$\vec{J}_I$ plane is denoted by $\vec{V}_{Des,temp,xy}$ and is used with the desired vertical velocity in the inertial frame, $\dot{z}_{Des}$, to compute the desired thrust vector by,

$$\vec{T}_{Des} = -K_p \ \vec{e}_{xyz} - K_d \left(\vec{V}_P - \left(\vec{V}_{Des,temp,xy} + \dot{z}_{Des} \ \vec{K}_I \right) \right) + \vec{W} - \vec{F}_{L,D,S} \tag{26}$$

Combining the weighted velocity error in this way allows the desired velocity vector to point in the direction of the desired final position in the horizontal plane, while still maintaining traditional proportional-derivative control for altitude. Feeding forward terms

for the weight, lift and drag, or at least bias values as midrange, static, placeholder replacements, the collective desired thrust is formed in Equation 26. This desired thrust vector computed from the outer loop is then provided to the inner loop to find desired moments on the body necessary to maintain stability.

Two euler angles, γ_i and β_i, are used to represent roll and pitch rotation angles from the payload frame to the *ith* vehicle's frame. This is achieved by means of actuated gimbals between the attach points and vehicles. This frame is obtained by rotating about the $\vec{J}_P$ axis by angle γ_i, and then the resulting *x* axis by β_i. The resulting rotation matrix is given by $R_{P\to Vi,pre}$, and vehicle frame unit vectors are given by $\vec{I}_{Vi}$, $\vec{J}_{Vi}$, and $\vec{K}_{Vi}$. For the remainder of this section, the *i* subscript is dropped for simplicity. The desired rotation angles from the payload to the vehicle are computed on a per-vehicle basis and are given by,

$$\begin{bmatrix}\beta_{pre}\\ \gamma_{pre}\end{bmatrix} = \begin{bmatrix}\sin^{-1}\left(\vec{T}_{Des}\cdot\vec{J}_P \,/\, \|\vec{T}_{Des}\|\right)\\ atan\left(\left(-\vec{T}_{Des}\cdot\vec{I}_P\right)/\left(-\vec{T}_{Des}\cdot\vec{K}_P\right)\right)\end{bmatrix} \tag{27}$$

Here the subscript *pre* denotes the preliminary angles, as these are used in the first step to give an initial rotation. Further, the negative signs in the top and bottom of the arctangent calculation are significant when calculating the quadrant aware arctangent.

The inner loop of this control law, Equation 29, results in commanded payload-frame moment commands L_{Des}, M_{Des}, and N_{Des} and comprise the $\vec{M}_{Des}$ vector components (and are used to achieve the desired orientation). The orientation error vector, $\vec{e}_{ori}$ shown in Equation 28, is derived by the vee map from the skew symmetric matrix produced from the error in rotations,

$$\vec{e}_{ori} = 0.5\left(R_{I\to P_{Des}}\, R_{I\to P}{}^{T} - R_{I\to P}\, R_{I\to P_{Des}}^{T}\right)^{\vee} \tag{28}$$

$$\vec{M}_{Des} = -K_{p,in}\,\vec{e}_{ori} - K_{d,in}\,\vec{e}_{\Omega} \tag{29}$$

where $R_{I\to P_{Des}}$ is a desired payload rotation matrix created from roll, pitch, and yaw angles stipulated by the user. The value for $\vec{e}_{\Omega}$ is computed similarly to Equation 23 but uses the payload's true and desired angular velocities. The diagonal matrices, $K_{p,in}$ and $K_{d,in}$, are the proportional and derivative gains for the inner loop. The rotation matrix $R_{I\to P_{Des}}$ can be multiplied by $R_{P\to Vi,pre}$ to generate a preliminary full rotation matrix from the inertial frame to the desired vehicle frame, $R_{I\to Vi,pre}$.

A unit direction thrust vector is created for the preliminary vehicle thrust command in the payload frame. Because all the vehicles are assigned the same preliminary β_{pre} and γ_{pre} gimbal angles in this first step, only one thrust vector direction is necessary for all vehicles, hence Equation 30 does not use the subscript *i* on $\vec{K}_{V,pre}$. The incurred, unit moment vector acting on the payload, caused by the vectored thrust from each of the attached vehicles, is given by,

$$\vec{M}_{dir,Vi,pre} = \vec{r}_{\oplus P\to Vi} \times \vec{K}_{V,pre} \tag{30}$$

The magnitude of the vehicle radius vectors projected onto the $\vec{I}_P$ - $\vec{J}_P$ plane is used to define a desired yaw direction vector, $\vec{N}_{dir,Des}$, in Equations 31 and 32. This unit vector is perpendicular to the radius vector, and it will be in the direction of this vector that a

corrective delta thrust will be created to solve the second step of the control allocation scheme.

$$mag_{r,Vi} = \|P_P \, \vec{r}_{\oplus P \to Vi}\| \tag{31}$$

$$\vec{N}_{dir,Des,i} = \left(\vec{r}_{\oplus P \to Vi} \times \vec{K}_P\right) / \, mag_{r,Vi} \tag{32}$$

The first step of the control allocation problem is completed by computing a preliminary thrust magnitude command for each vehicle, *Vi*, in the payload frame, $T_{Vi,pre}$, which acts in the direction of the gimbal prescribed by β_{pre} and γ_{pre}. This is done by using a Moore-Penrose pseudoinverse to solve Equation 33, which resolves the desired roll and pitch commands, as well as the thrust vector prescribed by the outer loop.

$$\begin{bmatrix} T_{mag} \\ L_{Des} \\ M_{Des} \end{bmatrix} = \sum_{i=1}^{n} \left[\begin{bmatrix} 1 \\ P_P \, \vec{M}_{dir,Vi,pre} \end{bmatrix} T_{Vi,pre} \right] \tag{33}$$

In Equation 33, the thrust magnitude, T_{mag}, is given in Equation 34,

$$T_{mag} = \vec{K}_{V,pre} \cdot \vec{T}_{Des} \tag{34}$$

Each vehicle now has a commanded, preliminary thrust vector parallel to $\vec{T}_{Des}$, and the first step of the control allocation is complete. When the radius vectors are crossed with these preliminary thrusts, a resultant yaw torque will act on the payload since no information was included in Equation 33 to account for any desired yaw moment. Because of this, any residual yaw inadvertently produced by these vehicle thrusts is subtracted from the total desired yaw produced by the control law in Equation 29. The second step in the

control allocation method computes corrected vehicle rotation angles that achieve the desired yaw with no effect on body roll and pitch moments already resolved by Equation 33. This is done by enforcing that a change in thrust, $\Delta\vec{T}_{Vi}$, lies only in the payload $\vec{I}_P - \vec{J}_P$ plane, and achieves maximum effectiveness for each vehicle by assigning it in the direction described by $\vec{N}_{dir,Des,i}$. The magnitude of this correction is thus minimized by directing it along this vector that is perpendicular to the radius vector.

Given a preliminary thrust magnitude assignment for each vehicle from Equation 33, the "residual" yaw, N_{res}, is defined by,

$$N_{res} = N_{Des} - \sum_{i=1}^{n} \left[\left(\vec{M}_{dir,Vi,pre} \cdot \vec{K}_P \right) T_{Vi,pre} \right] \tag{35}$$

In this equation, the yaw moment caused by the rotation of the vehicles with respect to the payload is subtracted from the desired yaw moment from the inner loop control law. Now, in Equation 37, N_{res} is divided over the *n* vehicles attached to the payload. The vector form of N_{res}, in Equation 36, is constructed so as to not introduce any additional pitch or roll moments on the composite vehicle when Equation 37 is used to solve for the thrust correction, ΔT_{Vi}.

$$\vec{N}_{res} = N_{res}\, \vec{K}_P \tag{36}$$

$$\vec{N}_{res} = \sum_{i=1}^{n} \left[\left(\vec{r}_{\oplus P \to Vi} \times \vec{N}_{dir,Des,i} \right) \Delta T_{Vi} \right] \tag{37}$$

Using the matrix inverse if *n*=3, Equations 33 and 37 can be solved for unique $T_{Vi,pre}$ and ΔT_{Vi} values. However, when considering more than three vehicles, Equations

33 and 37 are underdetermined systems that can be solved for a minimum-norm solution using the Moore-Penrose pseudoinverse.

The preliminary thrust and individual vehicle delta thrusts are summed into a new desired thrust vector required for each vehicle, in Equation 38.

$$\vec{T}_{Vi,final} = \Delta T_{Vi}\, \vec{N}_{dir,Des} + T_{Vi,pre}\, \vec{K}_{V,pre} \tag{38}$$

Applying Equation 27 to the new desired thrust vector, $\vec{T}_{Vi,final}$, results in a new set of γ_i and β_i for each vehicle. These are the final commanded vehicle orientation angles, which are assumed to be achievable instantaneously (quasi-statically) at the given control time step. Equation 39 completes the second step of the control allocation problem, relating how each vehicle resolves its own commanded angular velocities, ω. The angular velocities for the individual rotors of the quadrotor are found using a given thrust coefficient, c_T, (since the motor angular velocity is the actual control signal applied to the motors). The quadratic relationship between thrust and rotor angular velocity is proposed and justified in [3].

$$\left\|\vec{T}_{Vi,final}\right\| = \sum_{j=1}^{m} \left[c_T\, {\omega_{Vi,j}}^2\right] \tag{39}$$

In addition to the above flight controller, an algorithm may be added to the overall control scheme to set the angle of attack of the payload in forward flight. Consider the use of an airfoil-shaped payload with known lift and drag characteristics as a function of angle of attack. This control law seeks to adjust angle of attack so that the lift of the payload and thrust of the vehicles exactly equals the weight of the composite payload (if possible). Given an initial angle of attack set point, α_{init}, the controller uses proportional and

derivative scalar gains, $k_{p,\alpha}$ and $k_{d,\alpha}$, to compute an angle of attack perturbation $\Delta\alpha$ given by,

$$\Delta\alpha = -k_{p,\alpha}\, e_{Lift} - k_{d,\alpha}\, \dot{e}_{Lift} \tag{40}$$

where e_{Lift} is given by,

$$e_{Lift} = -\overrightarrow{W} \cdot \vec{K}_I - \vec{F}_{L,D,S} \cdot \vec{K}_I \tag{41}$$

and $\dot{e}_{Lift}$ is computed with the finite backward difference method.

3.2 Extended Kalman Filter for Adaptive Control

3.2.1 Overview

The equations describing the inner-loop PD controller and the allocation of the control signals show that the vectors from the payload center of gravity to each of the attached thrusters, or vehicles, are critical quantities that are needed in order to maintain proper attitude control of the system. These are called “radius vectors” herein. Due to the method’s reliance on these vectors in the control allocation scheme, an extended Kalman filter is used to estimate these radii. This allows the user to avoid the need to provide this information precisely to the system upon initialization. Removing the reliance on input data provided by the user would extend the usability of the control scheme to a much larger set of use cases. Instead of knowing precise measurements, the user could just select the number of vehicles attached to a payload, along with vehicle thrust and size data from the vehicle database. The system will then create its own geometric radius vector parameters

to use in the control allocation scheme as it stabilizes the aircraft after takeoff, then transports the payload to the destination using these self-generated parameters.

In order to minimize the number of computations and the complexity of the estimator, only the radius vectors are estimated (and the controller is therefore assumed to be robust to uncertainty in weight and inertia). This allows for better prediction of the torques being exerted on the system, as they are comprised of a cross product between the radius vectors and the respective vehicle thrusts. The thrust vectors are assumed to be known, calculated onboard by multiplying the rotor thrust coefficient with the motor commands sent from the controller at each time step. The mass, inertia, and drag parameters are not explicitly used in the control system, but the PD controller compensates for the lack of precise knowledge in these parameters via state feedback.

In order to compensate for lack of knowledge of other system parameters, nominal bias values are given to the controller. First, the *mg* term in Equation 4 is replaced by one half the total thrust capacity of all the vehicles attached to the payload. In this way, the controller in Equation 26 has to compensate for at most a mass error of double or half the expected value. Additionally, nominal values for the inertias and the coefficients of drag are also left as static values. Finally, the randomization of the payload CG location and the vehicle attach points is combined into the estimated radius vectors, reducing the number of parameters to be estimated by half.

3.2.2 Mathematical Development

The formulation of the radius vector EKF follows closely to that used in [25]. The implementation of the extended Kalman filter operates on an augmented state vector in

Equation 42. The augmented state is the combination of the thirteen-element state vector from the dynamic model with the radius vectors that will be estimated, as seen in Equations 43-44.

$$\vec{x}_A = \begin{bmatrix} \vec{x} \\ \vec{p} \end{bmatrix} \tag{42}$$

$$\vec{x} = [\, x\ y\ z\ q_0\ q_1\ q_2\ q_3\ u\ v\ w\ p\ q\ r\ ]^T \tag{43}$$

$$\vec{p} = [\, \cdots\ r_{Vi,x}\ r_{Vi,y}\ r_{Vi,z}\ \cdots\ ]^T \tag{44}$$

The extended Kalman filter uses an estimated, augmented state vector, $\vec{x}_{A,est}$, along with an incremental estimate vector, $\Delta\vec{x}_{A,est}$, to produce a new, updated, estimated state, $\vec{x}^{+}_{A,est}$, as seen in Equation 45.

$$\vec{x}^{+}_{A,est} = \vec{x}_{A,est} + \Delta\vec{x}_{A,est} \tag{45}$$

The incremental estimate is produced by multiplying the residual, i.e. the innovation, between the measured state and the estimated state by a Kalman gain, *K*, as in Equation 46.

$$\Delta\vec{x}_{A,est} = K\left(\vec{y}_m - \vec{h}\left(\vec{x}_{A,est}\right)\right) \tag{46}$$

For the simulation, a true, measured state, $\vec{y}_m$, is propagated using the dynamic model of the composite system, as described above. For this true, measured state, the parameters of the dynamic model use true values, i.e. values from the theoretical user's physical system. In contrast to this, the estimated state, $\vec{x}_{est}$, is created by replacing the true values with generic nominal values for all parameters except the radius vectors. As described above,

the mass, inertias, coefficients of drag, and vehicle yaw values are static, nominal values, in general not equal to the true values. Also, the model does not use the mass center location nor the attach point locations directly. Only the vector between these points, described here as radius vectors, are estimated.

Simultaneous to the RK4 propagation of the estimated state, the covariance matrix of the augmented state error, *P,* is integrated. The dynamics of this covariance matrix are described by the Riccati equation, in Equation 47.

$$\dot{P}(t) = F(\vec{x}_A, t)P(t) + P(t)F^T(\vec{x}_A, t) + G(t)Q(t)G^T(t) \tag{47}$$

The *F* matrix is the Jacobian of the augmented state dynamic model equations, and given by,

$$F(\vec{x}_A, t) \equiv \left.\frac{\partial \vec{f}}{\partial \vec{x}_A}\right|_{\vec{x}_A(t)} \tag{48}$$

All of the partial derivatives in the state Jacobian are derived analytically from the dynamic model equations of Chapter 2, with the exception of the drag inputs, which uses a forward finite difference derivative.

To create the Kalman gain, *K*, the Jacobian of the measurement equations, *H*, is needed, shown in Equation 49. Because the inertial measurement unit (IMU) is assumed to provide an output of the measured state directly, with no measurement of the radius vectors, the full state vector $\vec{x}$ effectively is the measurement equations, $\vec{h}$. This means the Jacobian

of the augmented state is the same as taking the Jacobian of the regular state, followed by a matrix of zeros, as in Equation 50.

$$H \equiv \left[\left. \frac{\partial \vec{h}}{\partial \vec{x}_A} \right|_{\vec{x}_{A,est}} \right] \tag{49}$$

$$\left[\left. \frac{\partial \vec{h}}{\partial \vec{x}_A} \right|_{\vec{x}_{A,est}} \right] = \left[\left. \frac{\partial \vec{h}}{\partial \vec{x}} \right|_{\vec{x}_{est}} \quad [0] \right] \tag{50}$$

The size of the zero matrix is the number of elements of the state vector by the number of elements to be estimated, in the $\vec{p}$ vector. Each vehicle has three elements in its radius vector, and there are n vehicles. Therefore, the zero matrix has thirteen rows and $3n$ columns. Finally, due to the measured state vector taking the place of the measurement equations, the Jacobian becomes the square identity matrix with thirteen rows, and is shown as,

$$\left. \frac{\partial \vec{h}}{\partial \vec{x}} \right|_{\vec{x}_{est}} = [I_{13\times 13}] \tag{51}$$

Assembling this together, the final H matrix, the Jacobian of the measurement equations, is static, and has thirteen rows, and 13+3n columns, as in Equation 52.

$$H \equiv \left[I \; [0] \right] \tag{52}$$

With the propagation of the estimated state and the covariance P complete, and the Jacobian of the measurement equations defined, we combine them along with the measurement covariance matrix, R, to compute the Kalman gain, K in Equation 53.

$$K = PH^T[HPH^T + R]^{-1} \tag{53}$$

Before repeating for the next time step, the covariance matrix P must be propagated. This is done by using the Kalman gain, and the measurement Jacobian, as in Equation 54, where I is the identity matrix.

$$P^+ = (I - K\,H)\,P \tag{54}$$

CHAPTER 4. SIMULATION RESULTS

Two overall sets of simulation results are provided in this work. The first set of simulation results studies three different scenarios to demonstrate the overall flight behavior of the modular vertical lift system, as well as to generate estimates for various performance characteristics including lift and power requirements. These cases assume perfect knowledge of vehicle locations with respect to the CG. First, an example scenario is provided in which four vehicles carry out a nominal payload delivery mission using an airfoil-shaped payload.

A second set of simulation results studies the stabilization characteristics of the system with uncertainty in system parameters. For these cases, comparisons are made for the controller's ability when the true attach points are symmetrically arranged about the payload CG versus asymmetrically arranged attach points. The stabilization performance of the system is shown for multiple example trajectories both with and without using the Kalman filter, demonstrating the benefit of performing real-time parameter estimation when the degree of parameter uncertainty is high.

4.1 Example Wing-Shaped Payload Delivery Mission

Simulation results are generated for an example scenario in order to demonstrate overall trends and performance of the modular vertical lift system. This scenario involves an airfoil-shaped payload weighing 150 pounds which is to be transported a distance of approximately 1100 m at a velocity of 23 m/s. Four attachment points are defined to be located at the four corners of the airfoil shaped, rigid container. It is assumed that the payload mass center is fixed and its position relative to the vehicles is known prior to flight, i.e. the radius vectors are given to the controller. Parameter values for vehicle and payload mass, inertial parameters, and drag parameters for this case were estimated from available experimental data, including the preliminary vehicle design described in the next section. For the purposes of drag and lift computations, the payload is modeled as a NACA0012 airfoil with a planform area of 2.75 m^2. All other vehicle parameters used in the simulation studies throughout this section are provided in the Appendix.

Figure 7 shows the position time histories of one of the four modular vehicles. Figure 8 and Figure 9 show the motor command time histories for a vehicle on the front versus a vehicle on the back of the payload, respectively (referencing the direction of flight). As shown in Figure 7, the vehicles arrive and attach to the payload around 25 seconds into the scenario. Once attached, the vehicles transport the payload a distance of about 1100 m over the course of 60 seconds, placing the payload on the ground. At this point the composite vehicle is at rest at the target location. At approximately 130 seconds into the scenario, the vehicles detach from the payload and return to their original starting location.

In Figure 8 and Figure 9, it is clear that much greater vehicle thrust, and hence high motor commands, are needed while the composite vehicle is accelerating (during the 42-47 second interval) and then thereafter while the composite vehicle is in steady-state flight or hover over the target, until 130 seconds. Around 90 seconds, the composite vehicle decelerates to a hover. After the vehicles land and detach from the payload around 130 seconds, a lower value of thrust is needed when the vehicles are in individual flight. The vehicles then travel back to their original positions, entering hover at about 180 seconds.

These figures, displaying the motor commands for forward and aft vehicles respectively, show a noticeable difference in motor commands while in forward flight. Figure 8 depicts the commands for a vehicle in front of the payload mass center, whereas Figure 9 depicts those of a vehicle to the rear. The thrust provided by the forward vehicles acts in a statically stable manner with respect to the payload, and hence the controller allocates greater commanded thrusts to these vehicles compared to the rear vehicles, which produce a destabilizing thrust moment. At about 85 seconds, the commanded rotor speeds are approximately 740 rad/s for the forward vehicle versus about 150 rad/s for the aft vehicle.

Figure 10 displays the angle of attack of the payload throughout the simulation time when the vehicles are attached. Once the composite vehicle reached the commanded travel altitude, the angle of attack controller started to command the optimal angle of attack for the given composite vehicle mass, and reached steady state at about 10 degrees. This was held from about 45 seconds when the vehicle reached its travel speed until just after 90 seconds when it began to slow down, nearing the delivery location. It is at this point the

angle of attack controller turns off, as the system is stopping and would not gain any advantage from maintaining an optimal payload angle of attack.

A final plot, Figure 11, shows the estimated energy consumption for the same two vehicles from Figure 8 and Figure 9 executing the example simulation trajectory. The power information required to obtain this figure is derived from energy consumption data taken from the prototype vehicle described in the Chapter 6. Figure 11 shows that the rear vehicle in this example expends about 35% less energy. This energy difference is caused by the thrust mismatch between the forward and aft vehicles in forward flight. This can be easily recognized by noting that the energy consumption slope is identical between the two vehicles in all but the forward flight portion of the scenario.

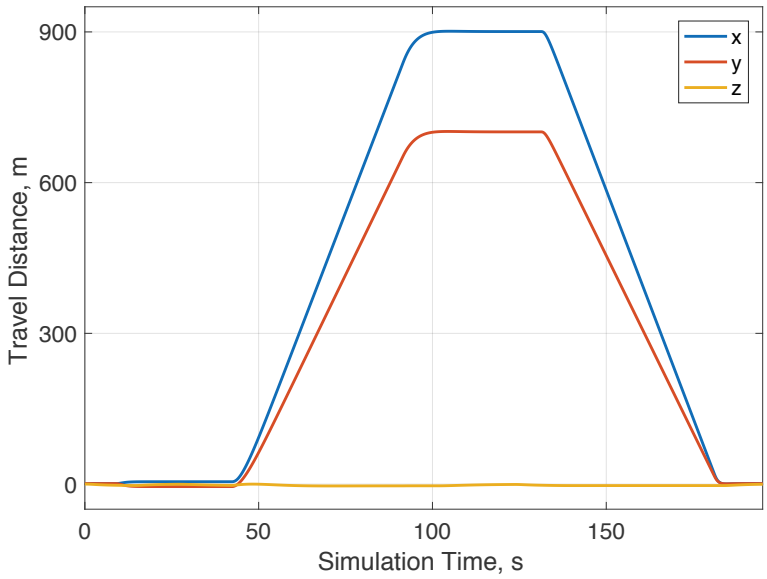

Figure 7 – Example Simulation Results. The vehicles first approach the payload, transport it 1100 meters, and finally return to their starting locations.

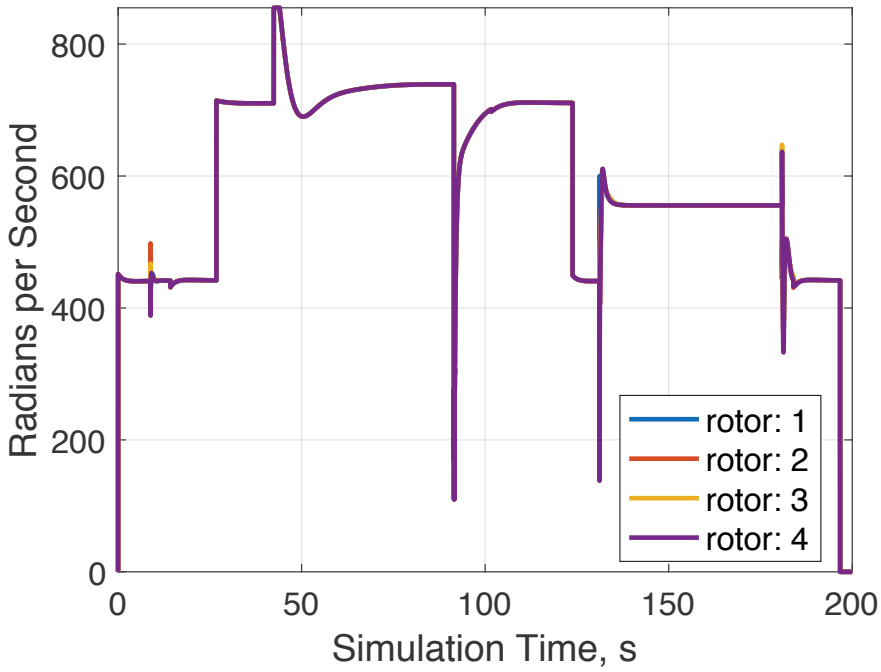

Figure 8 – Motor Command Histories for Example Simulation Results: Front Vehicle.

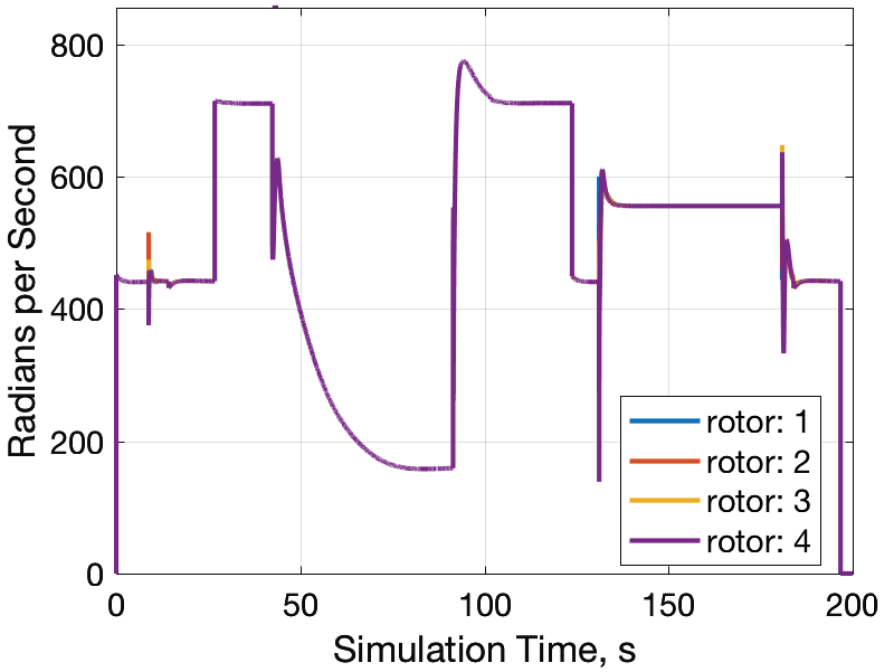

Figure 9 – Motor Command Histories for Example Simulation Results: Aft Vehicle.

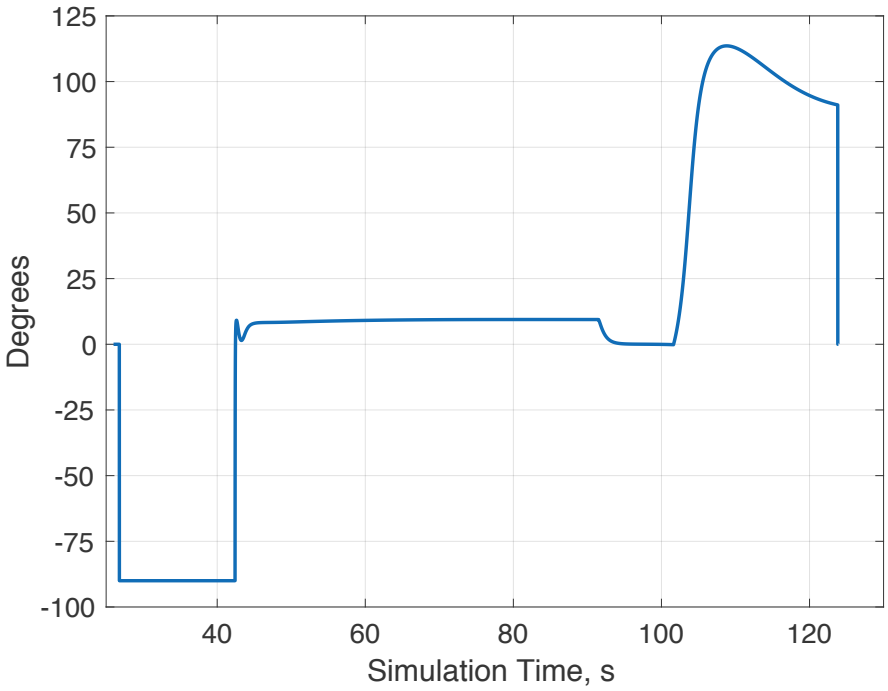

Figure 10 – Angle of Attack vs Time for Example Simulation Results.

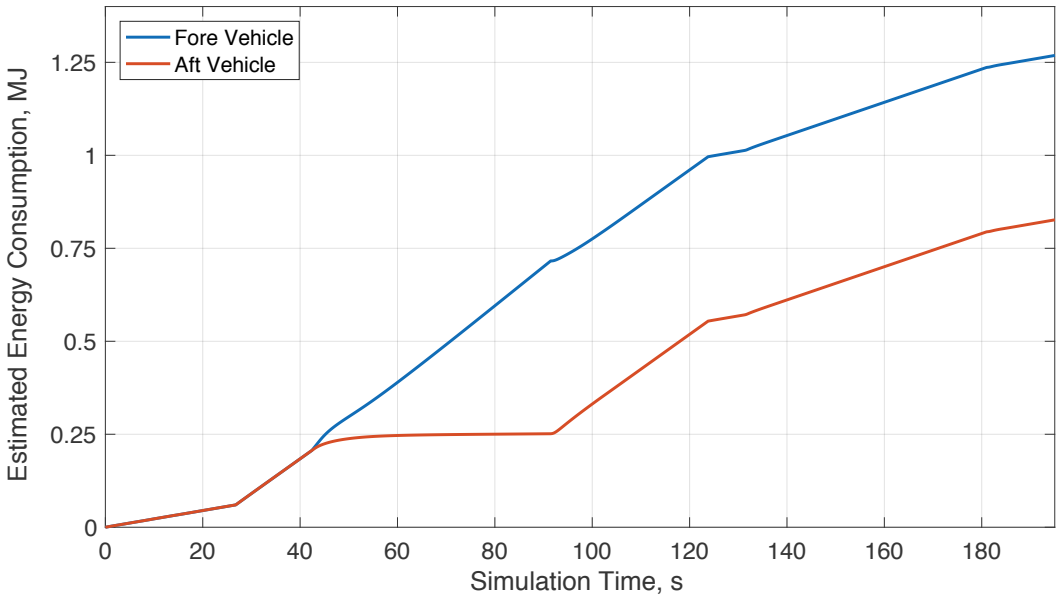

Figure 11 – Estimated Energy Consumption vs Time for Example Simulation Results.

4.2 Example Stabilization Trajectories

Several cases were simulated to test the capability of the controller in stabilizing a payload. Three types of situations were completed to show the difference in capability of the raw PID controller versus having a Kalman filter providing new, adapting input to the controller. For all cases, it is assumed the vehicles being used for transport are stock vehicles, with parameters and dynamics known to the system controller via some database. In all cases, the system is initialized from rest at some nominal altitude, and the controller tries to stabilize the system in a level hover condition. Further, in order to isolate the effects of radius vector estimate convergence on payload response, the system weight and inertias were set as static values. The controller has access to these values for each case assessed.

The first type of simulation involved assigning random values to the radius vectors, and allowing the controller to know precisely what these values are. This would be as if the user was able to measure the attach points and the location of the center of gravity on the payload to be transported prior to attaching the vehicles. If the user was able to upload these values to the controller, the control allocation algorithm will operate on perfect parameters throughout the mission. This gives an upper bound on what should be expected from the Kalman filter, since the filter can continuously improve the values used in the control allocation scheme.

The next type of simulation involved not giving any previous measurements to the controller. The controller is provided with randomized, erroneous parameter values and these are held constant throughout flight. This gives a lower bound for what should be

expected of the Kalman filter. If the filter is able to ascertain new, updated values, the control performance should improve beyond this static error case.

The last type of simulation involved activating the Kalman filter. With initial, randomized, erroneous parameter values used for the radius vectors, the filter is expected to improve these nominal radius vectors, reducing the error and aiding the PID controller in allocating control more effectively to the attached thrusters.

4.2.1 Example Stabilization Case with Full Knowledge

Figure 12 and Figure 13 show state time histories of an example stabilization case for a controller that has full knowledge of the composite vehicle parameters, and the attached vehicles are arranged in a symmetric pattern about the payload CG. The vehicle stabilizes almost instantaneously due to the perfect kinematics inversion that is performed during control allocation. All oscillations experienced while achieving steady-state are of very low amplitudes.

The motor history commands will tend towards saturation and induce loss of control sooner if the radius vector for any one of the vehicles is much shorter than any other radius vector. This result is visible in the asymmetric attach point scenario in Figure 14, where greater separation will occur with larger variations in radius vector lengths and placements. This behavior is what allows the Kalman filter to have better performance in some scenarios. As will be discussed below, a measurement that does not match an expected measurement due to a high motor command that saturates, the Kalman filter will adjust the radius vector to correct for the unexpected result, and saturation will not happen as often, leading to a larger number of cases that can be stabilized, as seen in Figure 40.

The gimbal histories in Figure 15, Figure 20, Figure 21, Figure 26, and Figure 27 are the commanded β and γ values for the gimbal rotations about the x and y axes, using a 2-1 rotation scheme. For the present case, the gimbal histories were identical, due to the near perfect kinematic inversion with correct radius vectors from the first time step.

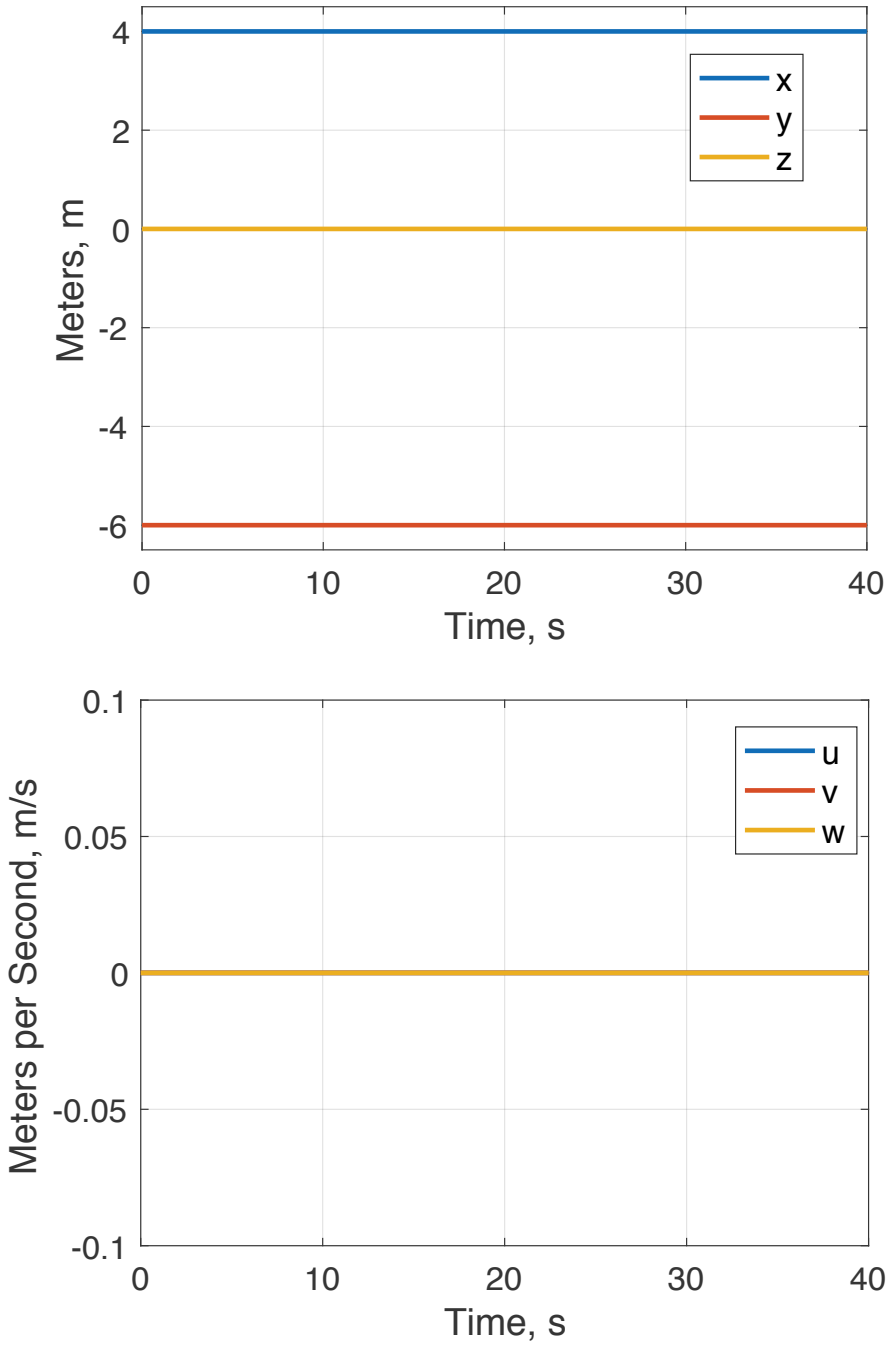

Figure 12 – Position & Velocity Stabilization Trajectories for Payload with Full Knowledge: Symmetric Attach Points.

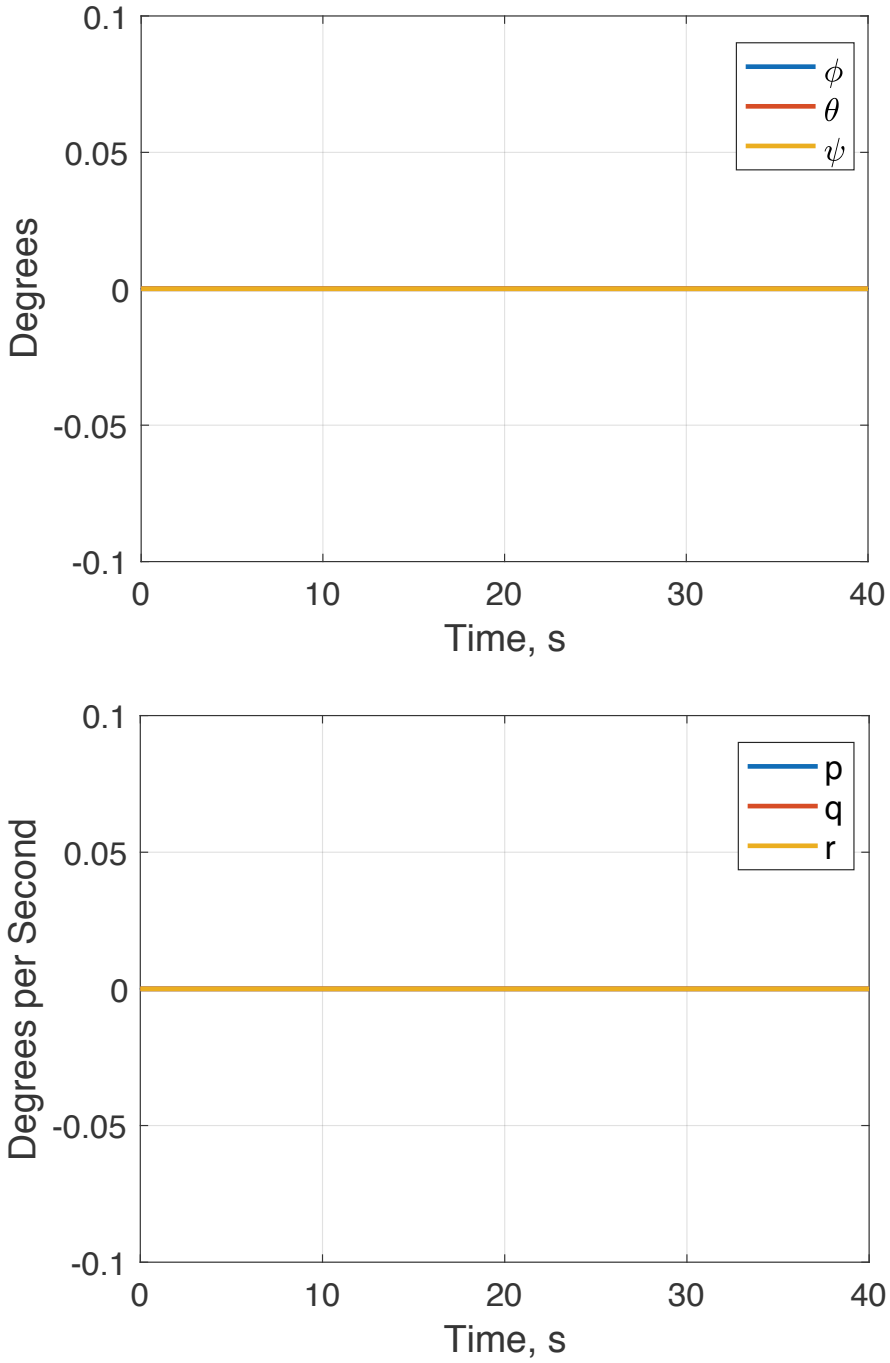

Figure 13 – Orientation & Angular Velocity Stabilization Trajectories for Payload with Full Knowledge: Symmetric Attach Points.

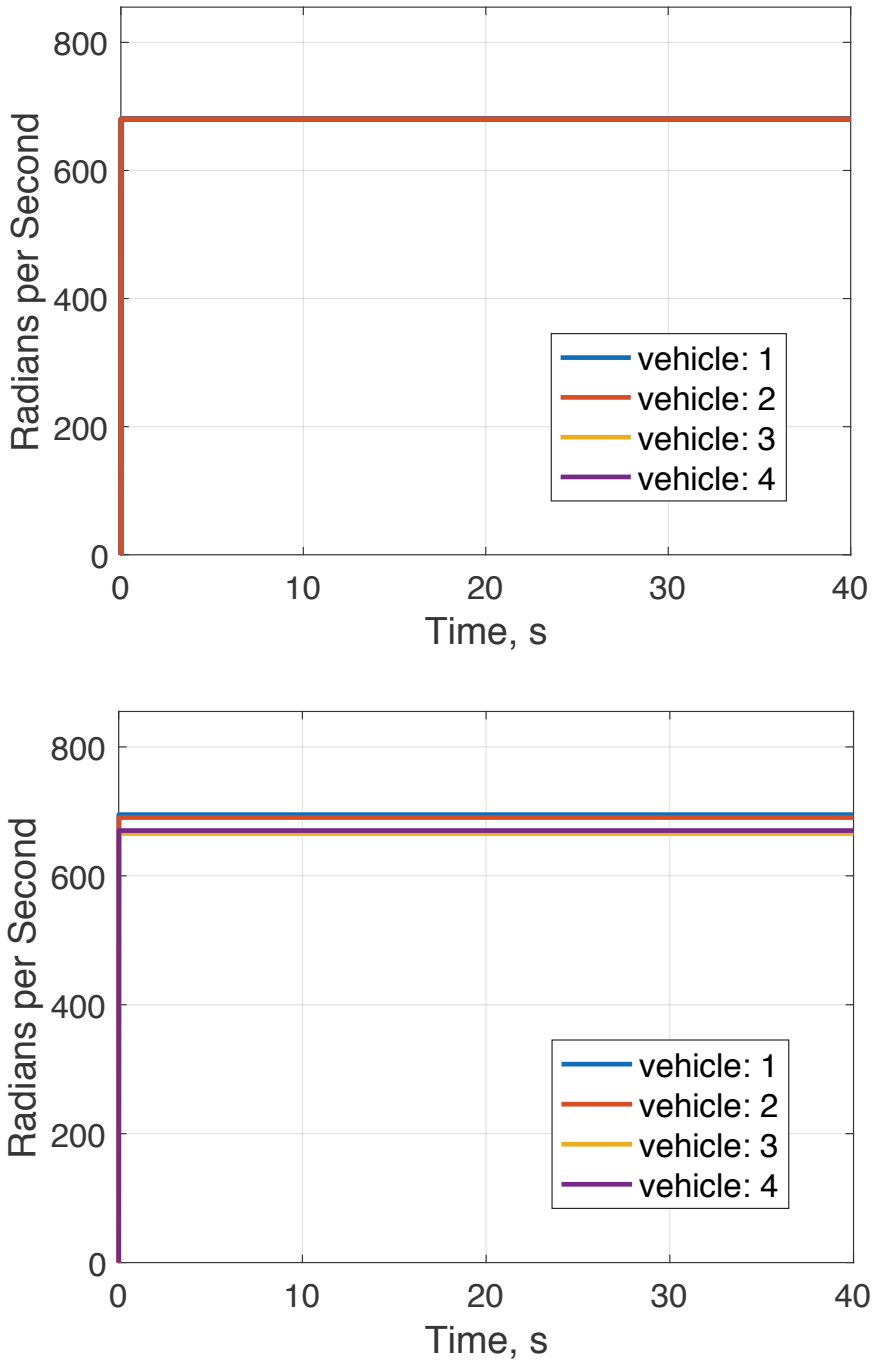

Figure 14 – Stabilization Motor Commands with Full Knowledge: Symmetric & Asymmetric Attach Points.

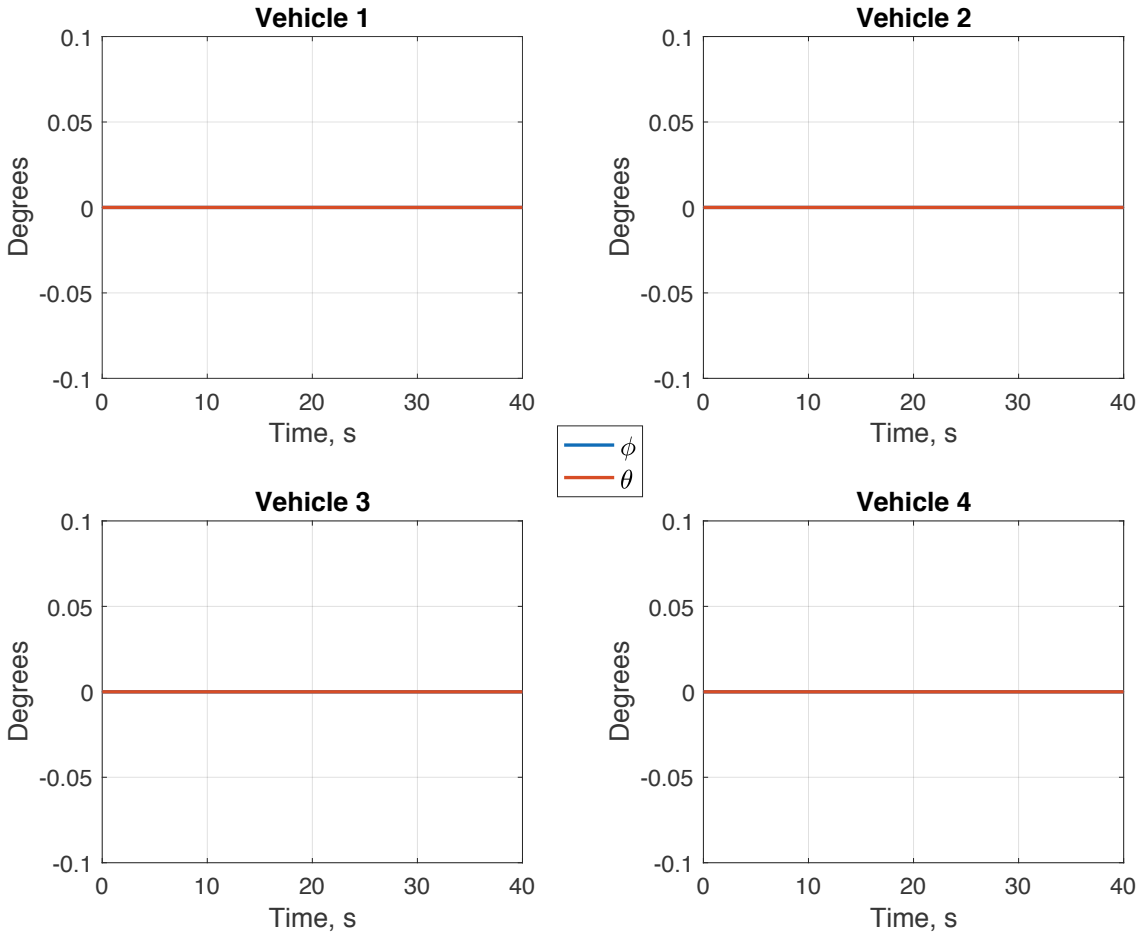

Figure 15 – Gimbal Histories for Full Knowledge Case: Symmetric Attach Points.

4.2.2 Example Stabilization Case with No Parameter Knowledge

Because the controller essentially treats the radius vectors as roll and pitch effectiveness values, the controller can stabilize the system effectively when the vehicles are arranged symmetrically about the CG. The controller only has to regulate the magnitude of the thrust to maintain correct altitude. When no knowledge of the parameter values is given to the controller, the PID can only compensate by changing the gimbal angles when the vehicles are attached in a non-symmetric pattern about the CG. Figure 16-Figure 19 show state time histories of the composite vehicle for symmetric and asymmetric configurations about the CG. As shown in Figure 19 and Figure 21, the payload pitches by about 17 degrees before the controller catches it by vectoring the thrusts from the vehicles. This correction happens very quickly – as shown in Figure 18 the altitude lost is relatively small.

This result has greater implications for the success of the Kalman filter. Due to no knowledge of the asymmetric parameters at the start, the resultant excitation in the payload dynamics as it rolls before settling at about 3 degrees roll and 17 degrees pitch gives ample information to the Kalman filter to quickly correct the radius vectors being used. The effect of this is shown in Figure 29, where the Kalman filter rapidly reduces the error of the radius vectors in the first half second, then the error levels off as the payload excitation reduces back to steady state.

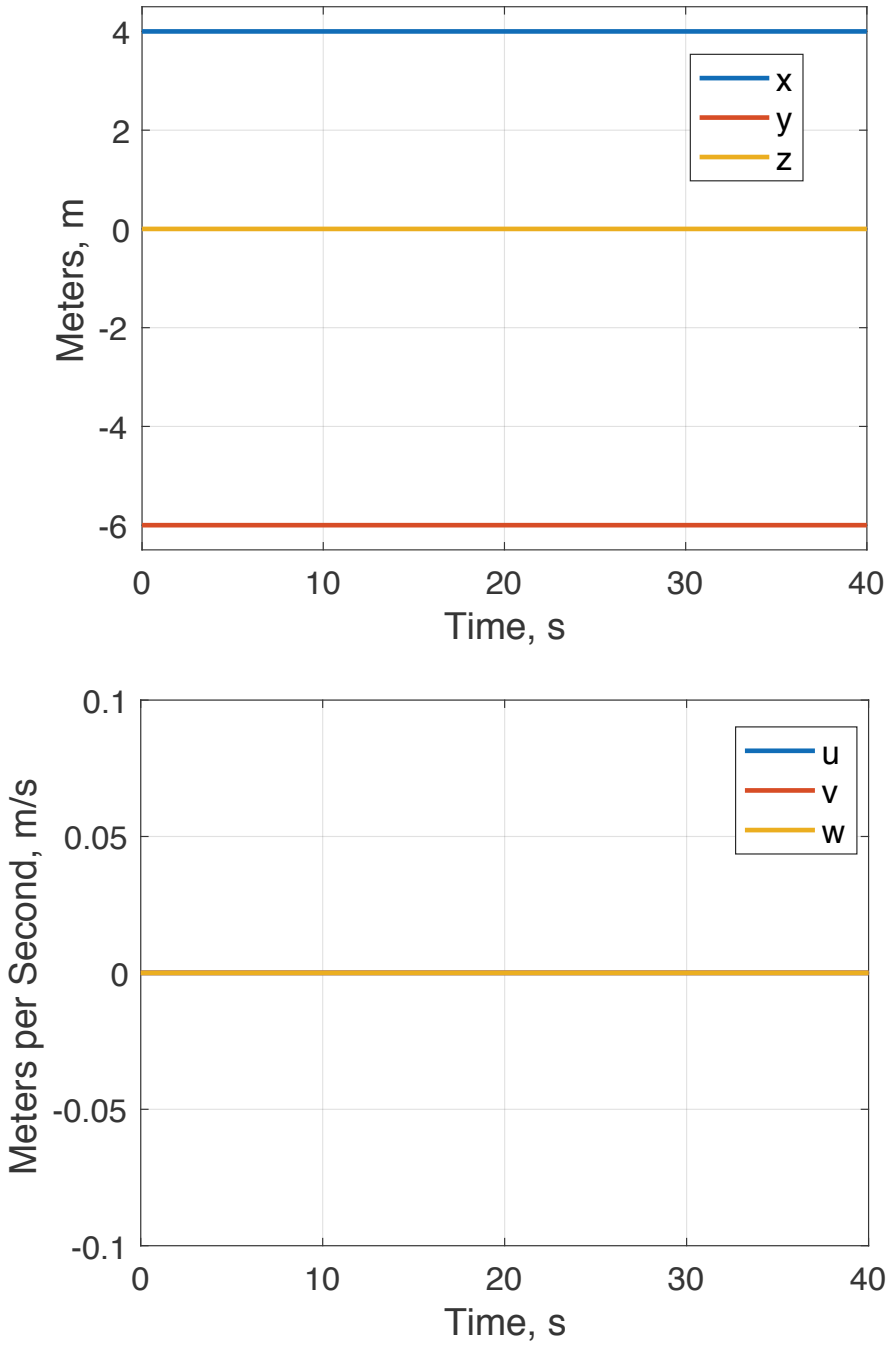

Figure 16 – Position & Velocity Stabilization Trajectories for Payload with No Knowledge: Symmetric Attach Points.

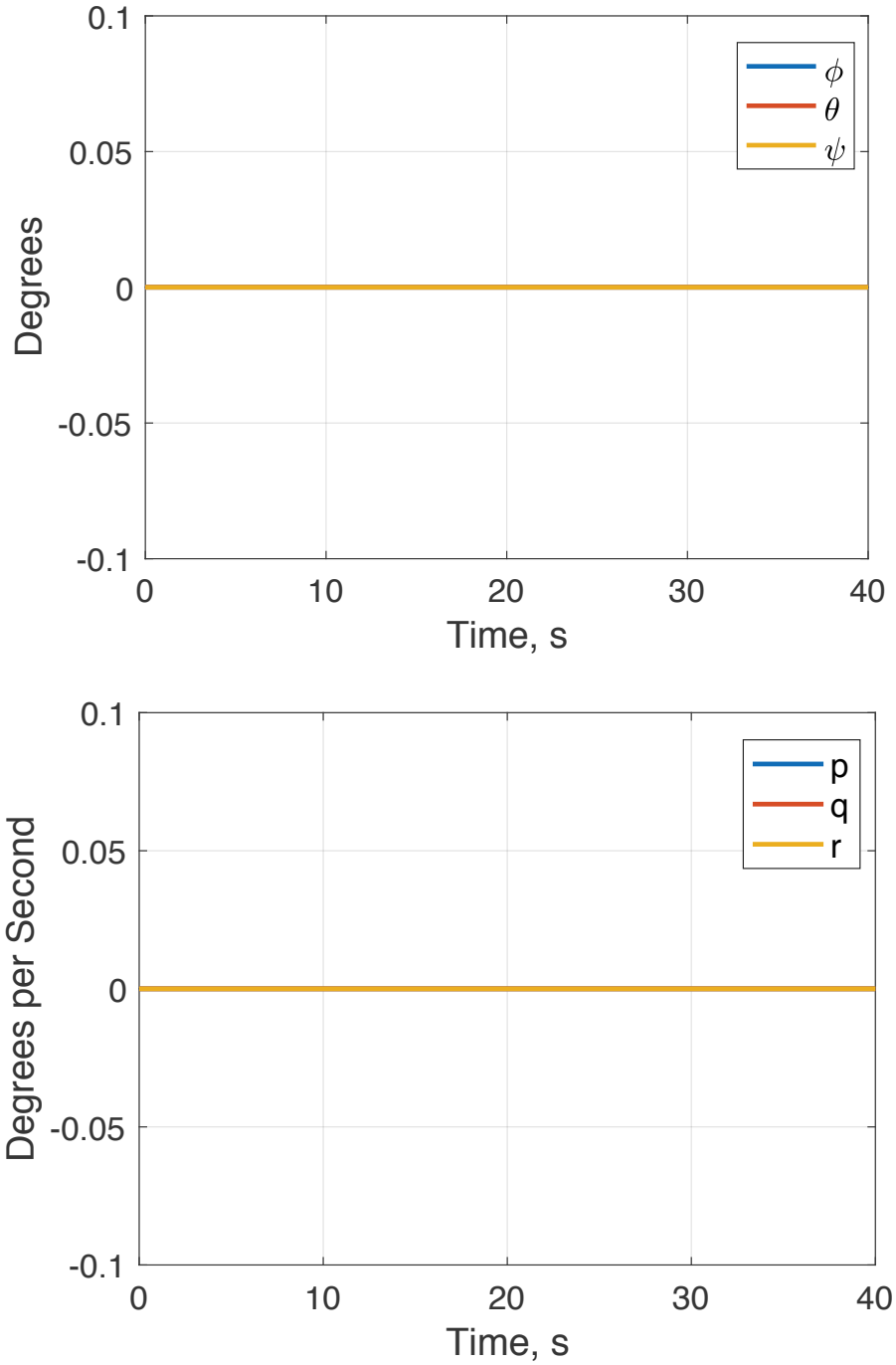

Figure 17 – Orientation & Angular Velocity Stabilization Trajectories for Payload with No Knowledge: Symmetric Attach Points.

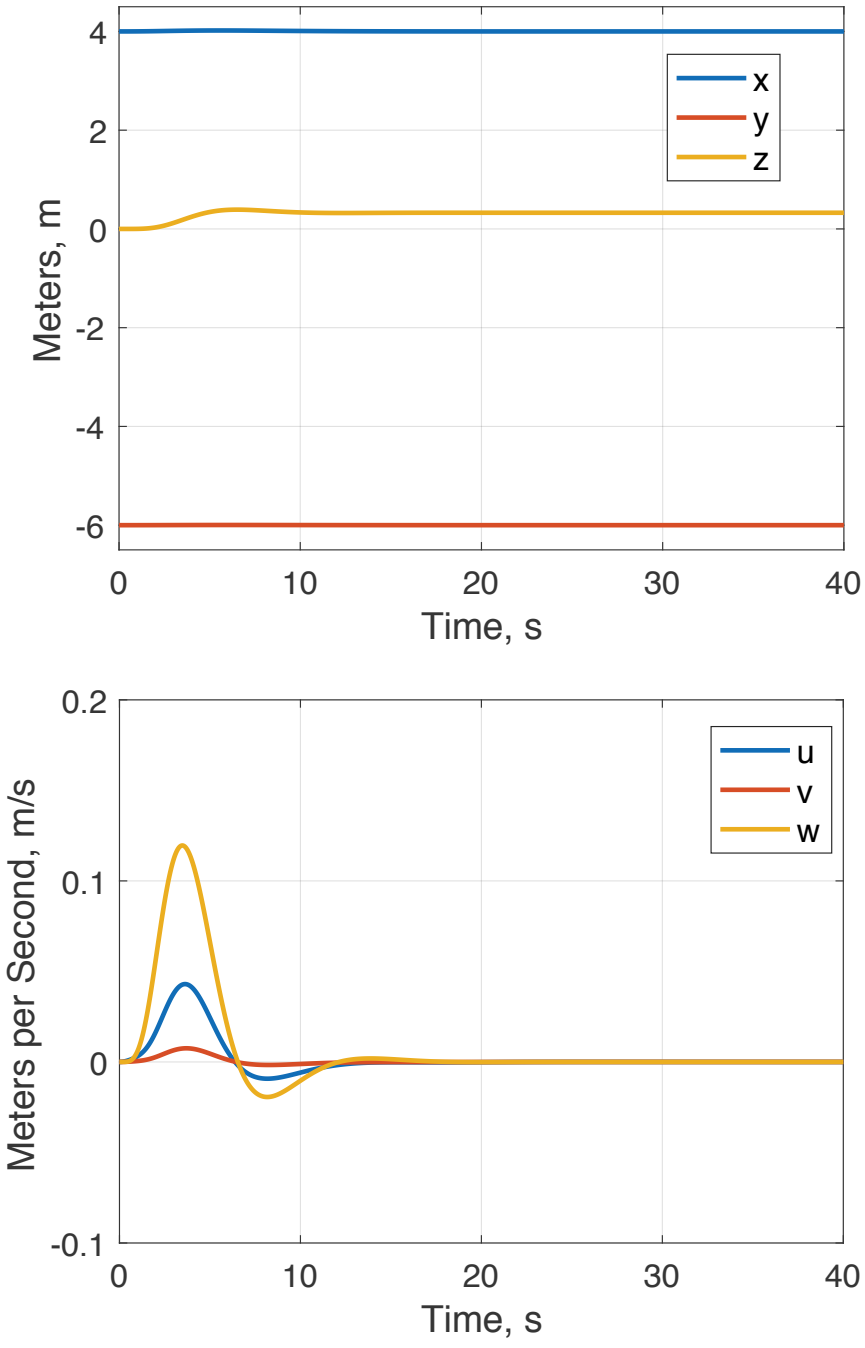

Figure 18 – Position & Velocity Stabilization Trajectories for Payload with No Knowledge: Asymmetric Attach Points.

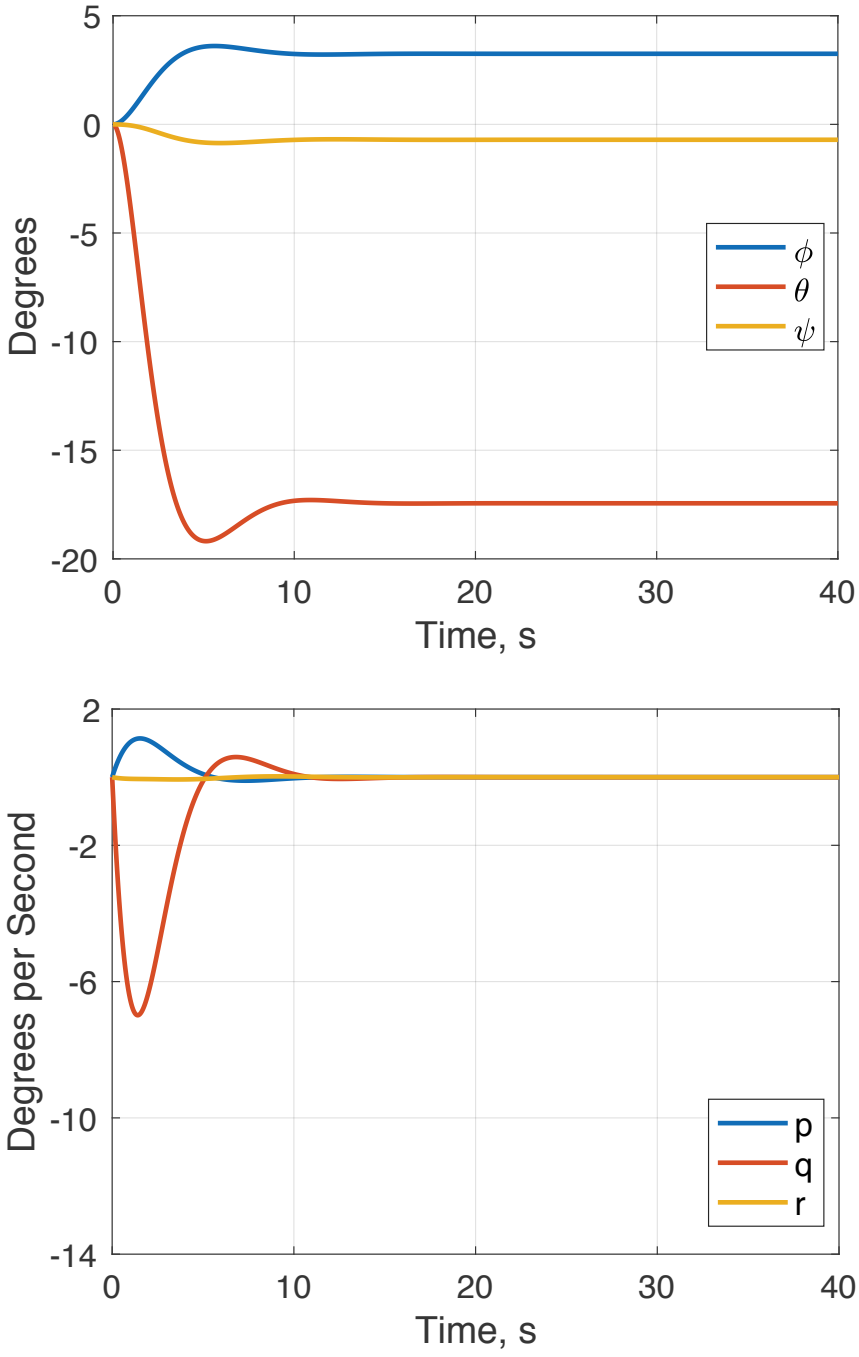

Figure 19 – Orientation & Angular Velocity Stabilization Trajectories for Payload with No Knowledge: Asymmetric Attach Points.

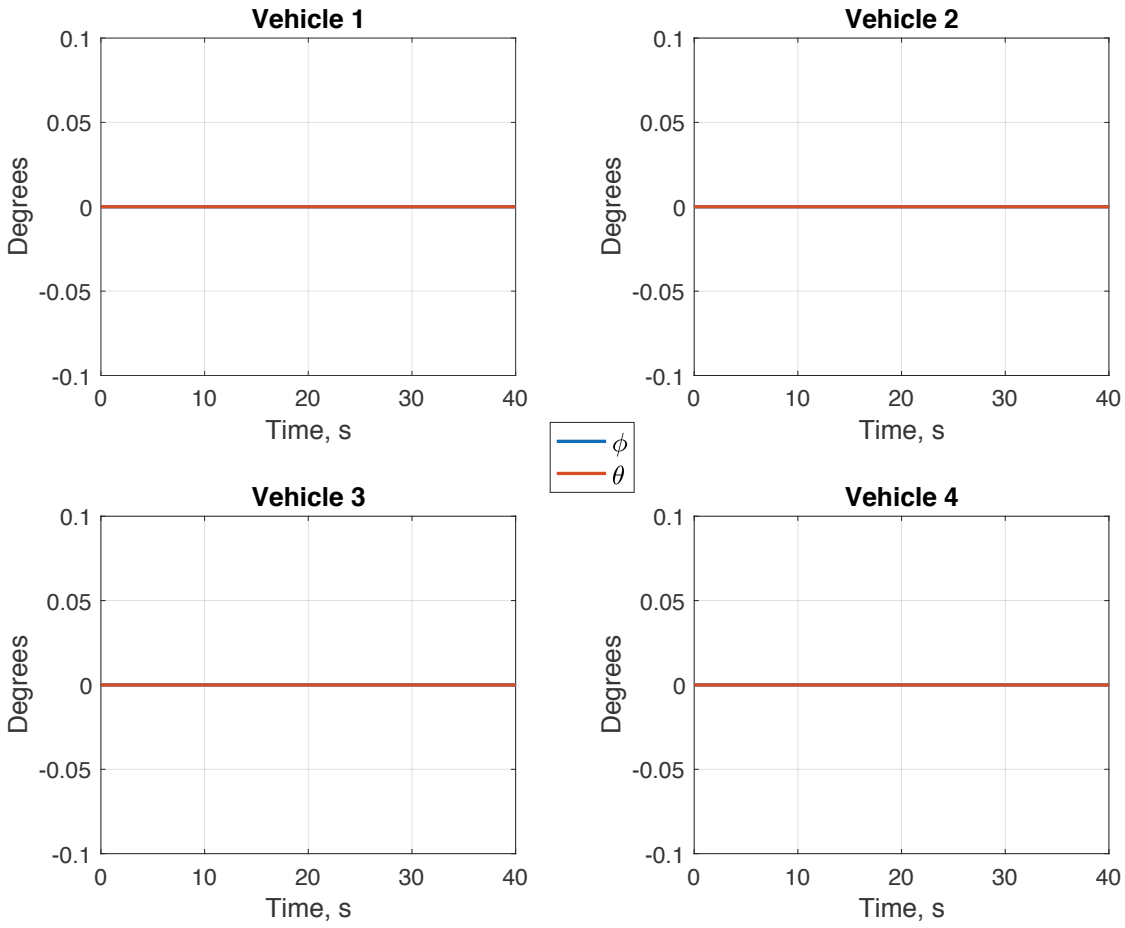

Figure 20 – Gimbal Histories for No Knowledge Case: Symmetric Attach Points.

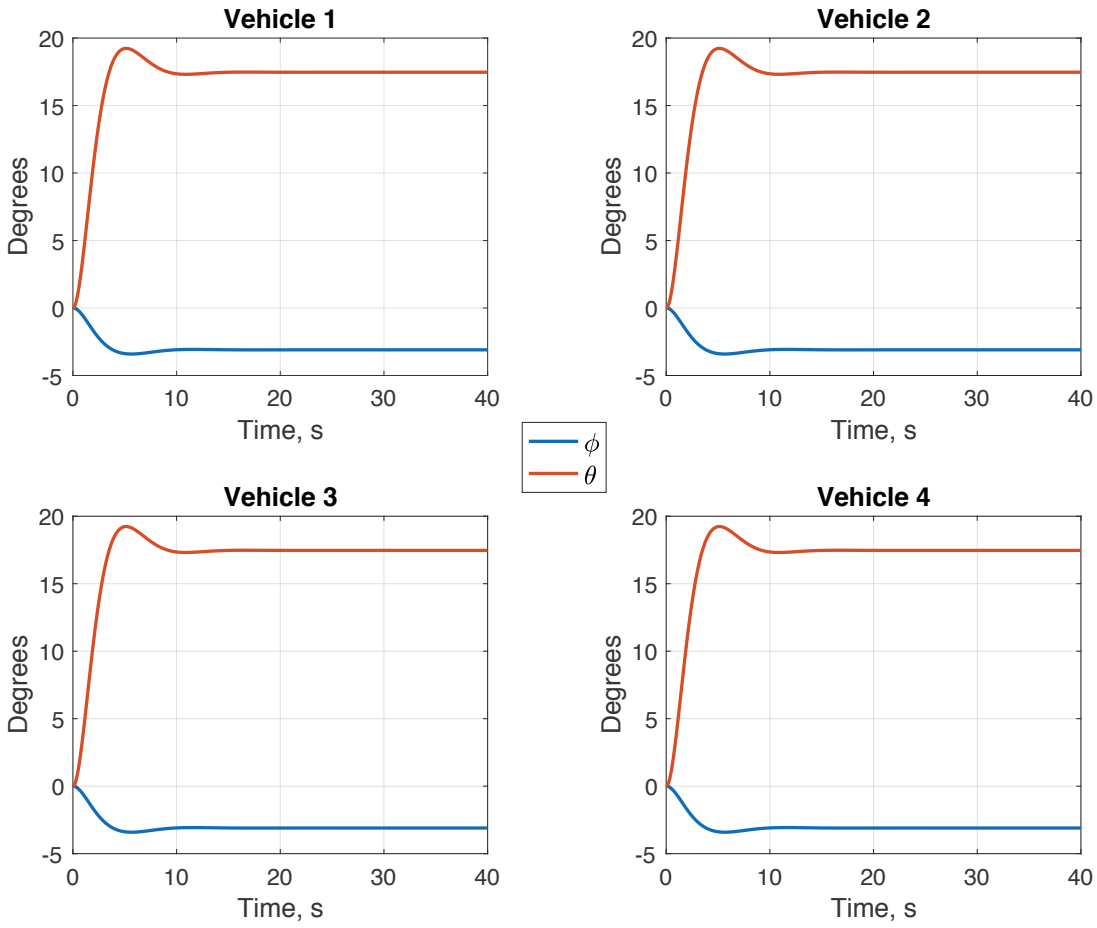

Figure 21 – Gimbal Histories for No Knowledge Case: Asymmetric Attach Points.

4.2.3 Example Stabilization Case with Active Kalman Filter

Figure 22-Figure 25 show stabilization trajectory time histories for both symmetric and non-symmetric configurations with the Kalman filter activated. Upon activating the Kalman filter, Figure 24 shows that the altitude control improves greatly over the case with no knowledge given to the controller. When a symmetric attachment scheme is used, the PID controller can stabilize the system quite effectively without updating the radius vectors, as seen in the previous section. As can be seen in Figure 29, the filter changes the parameters significantly with non-symmetric attach points. This is due to the inability of the PID to compensate quickly on its own, and the system starts to roll and pitch. Due to the gimballing, the excitation of the payload states remains relatively small, however. The thrusts can be vectored, and so the payload orientation can remain very close to zero while still traveling, unlike a normal quadrotor vehicle. The orientation perturbations that do occur allow the Kalman filter to operate on larger differences between the estimated state and the measured state, which in turn allows it to more quickly update the radius vectors as compared to the symmetric case. The controller still implements some gimballing, as seen in Figure 27, but it does not require as large, if any, steady-state gimbal angle to offset the skewed radius vectors, because the Kalman filter compensated by providing new corrected radius values.

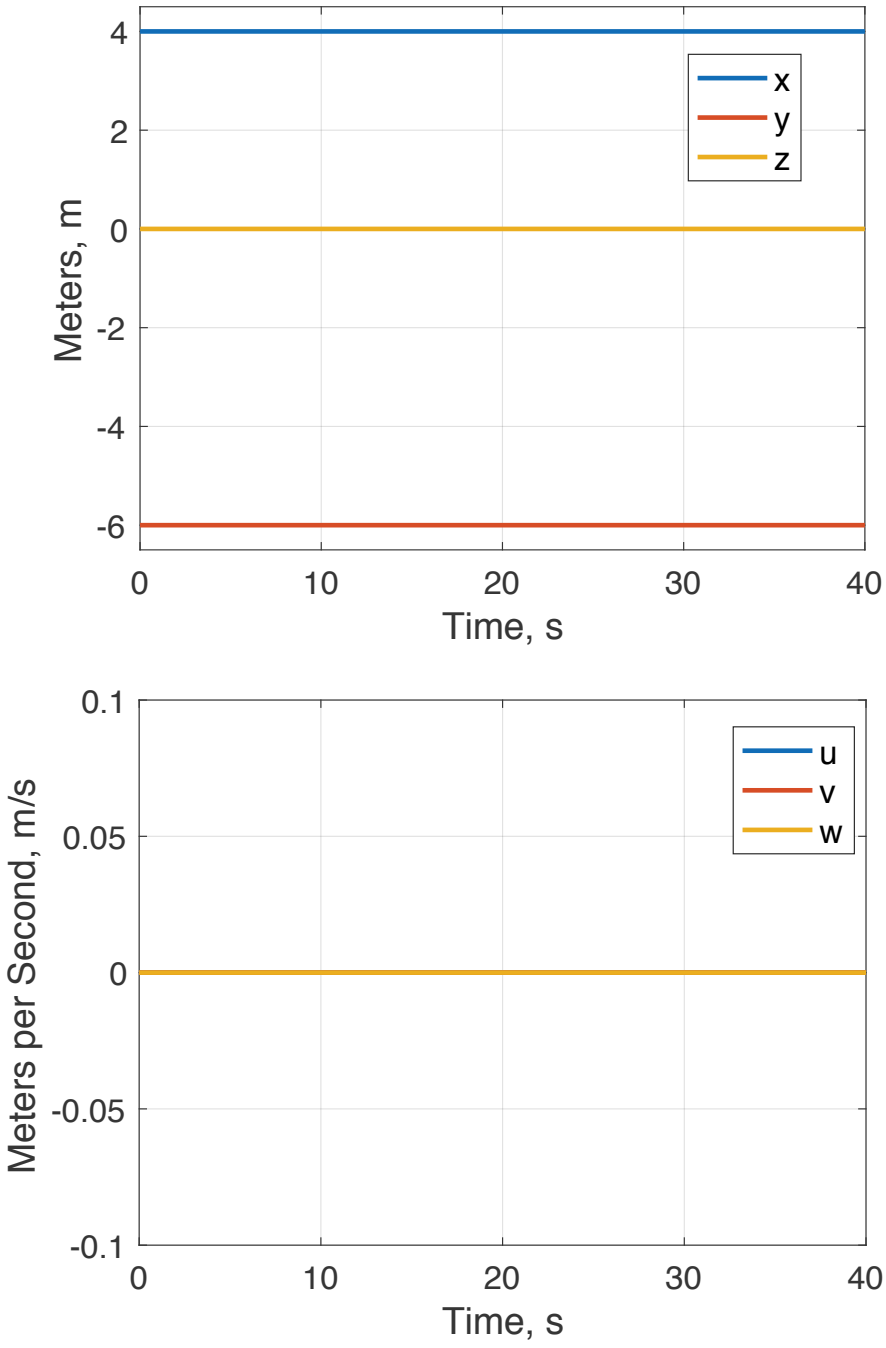

Figure 22 – Position & Velocity Stabilization Trajectories for Payload with Active Kalman Filter: Symmetric Attach Points.

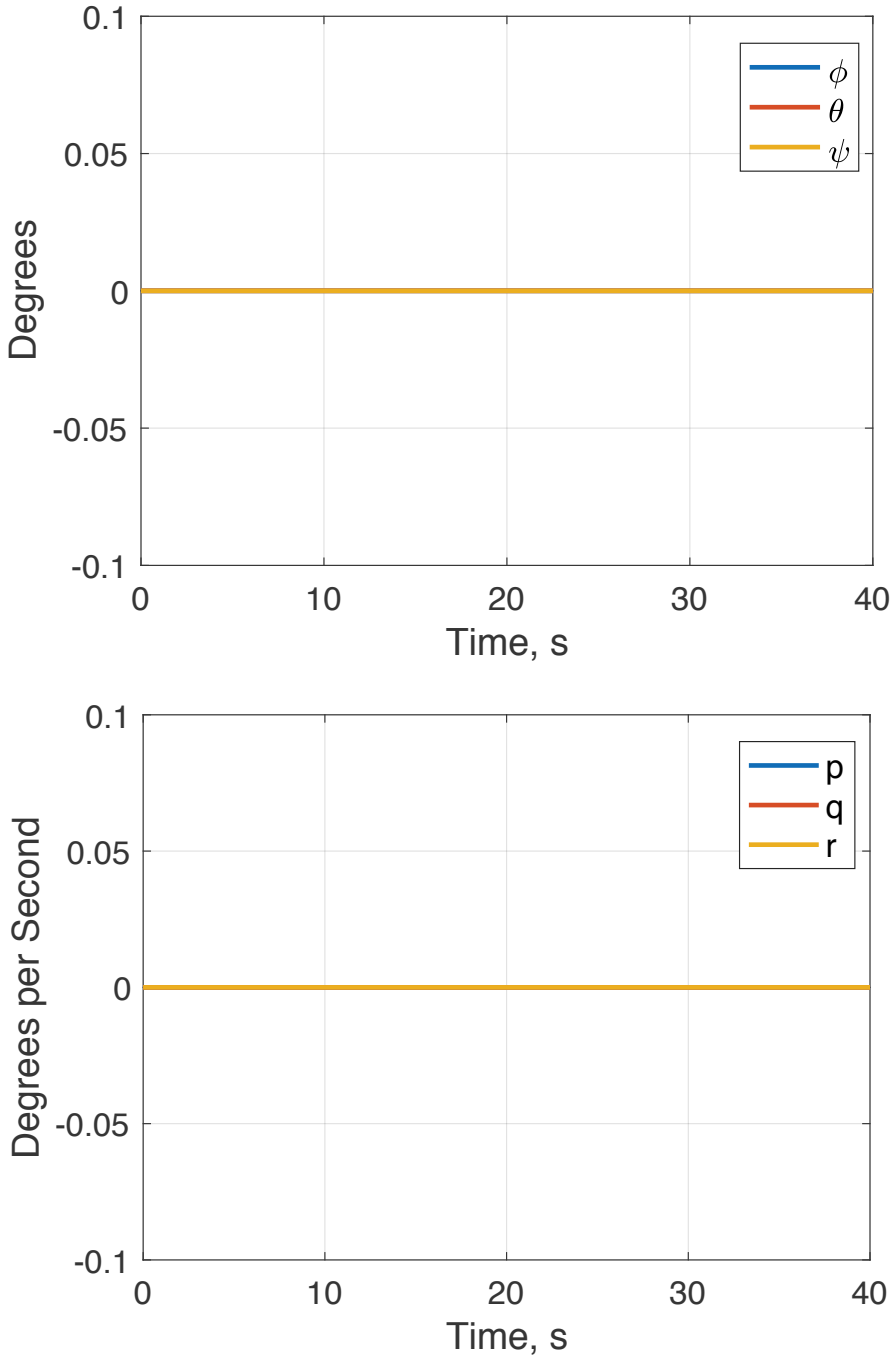

Figure 23 – Orientation & Angular Velocity Stabilization Trajectories for Payload with Active Kalman Filter: Symmetric Attach Points.

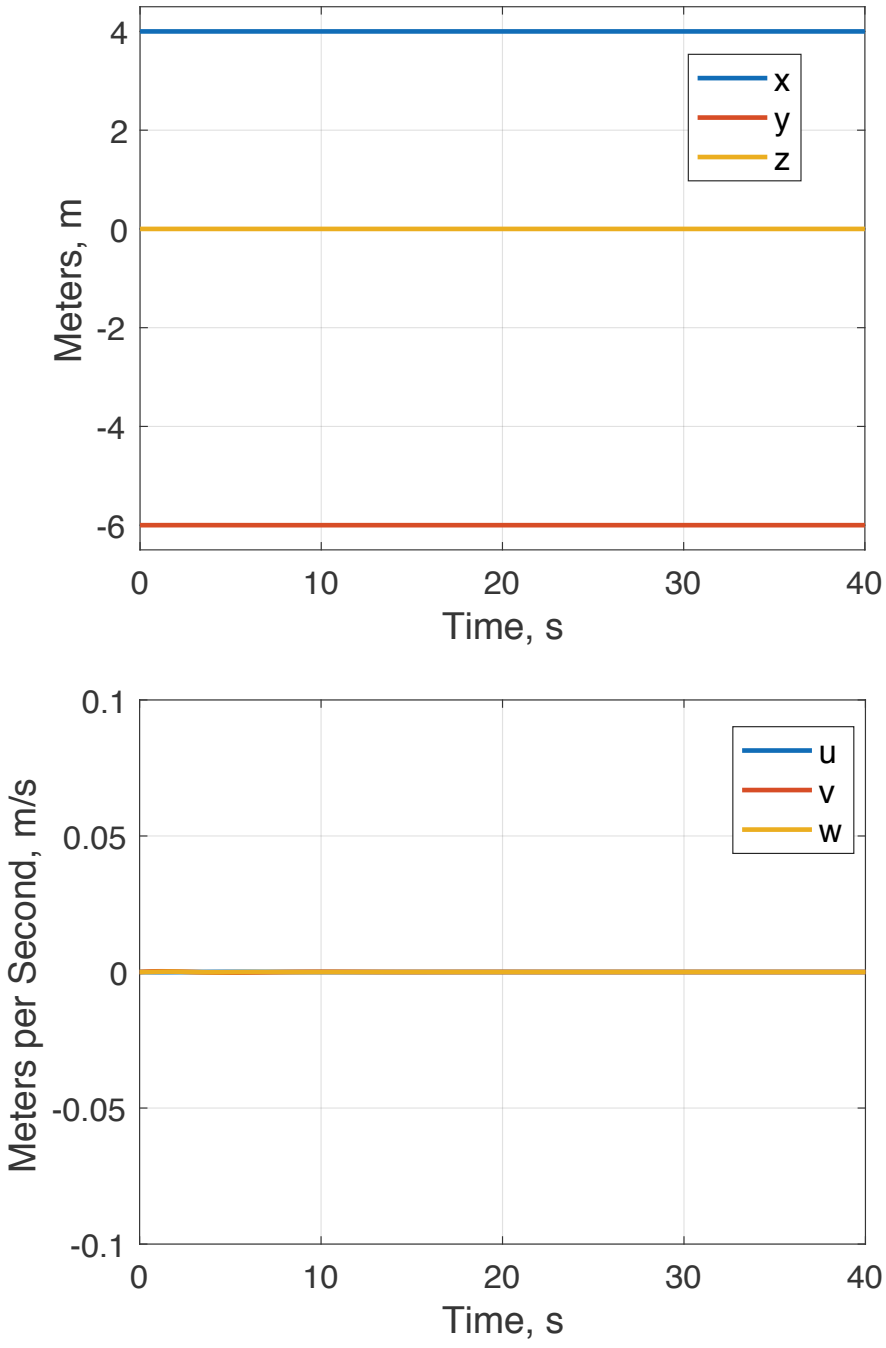

Figure 24 – Position & Velocity Stabilization Trajectories for Payload with Active Kalman Filter: Asymmetric Attach Points.

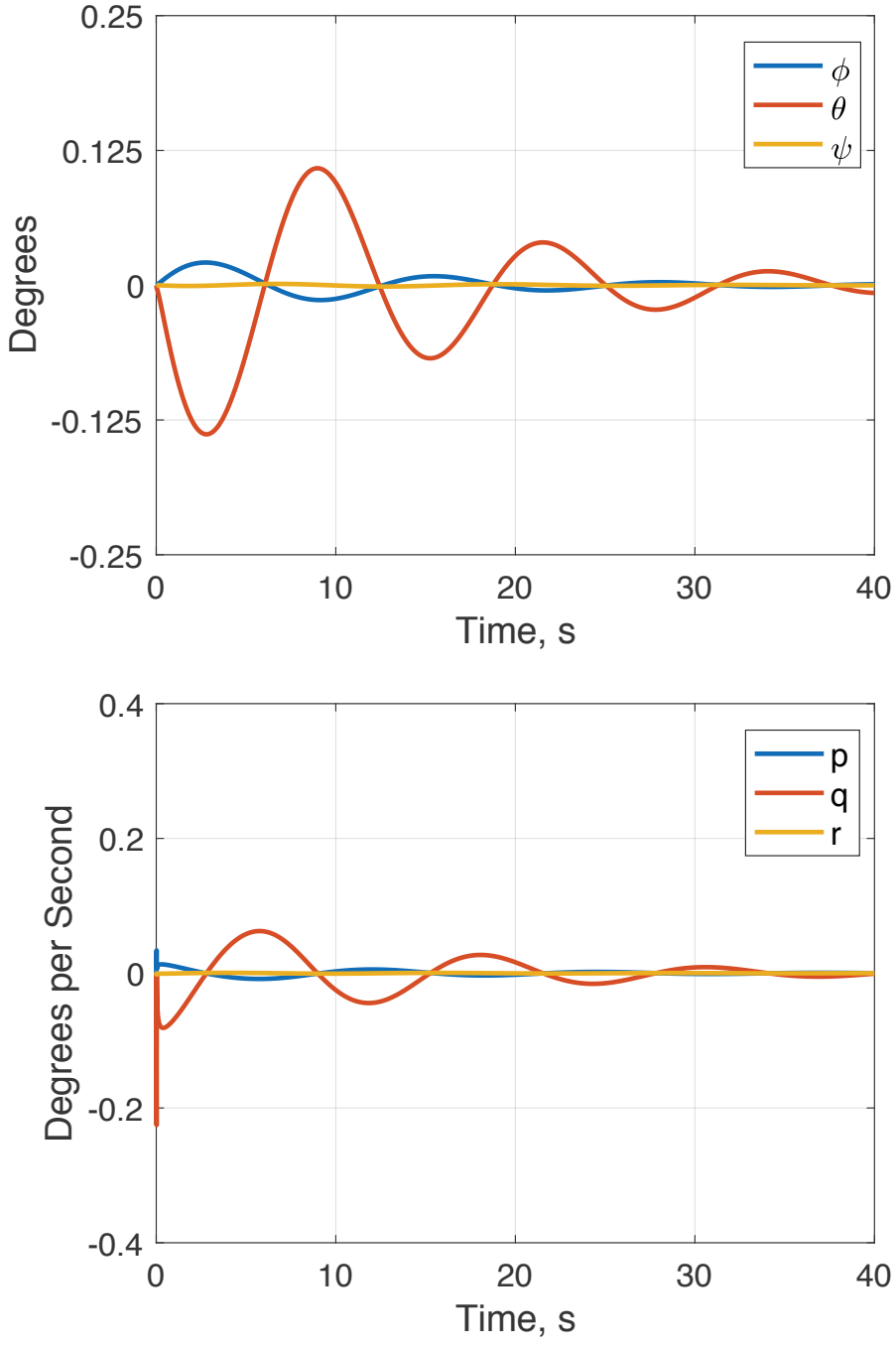

Figure 25 – Orientation & Angular Velocity Stabilization Trajectories for Payload with Active Kalman Filter: Asymmetric Attach Points.

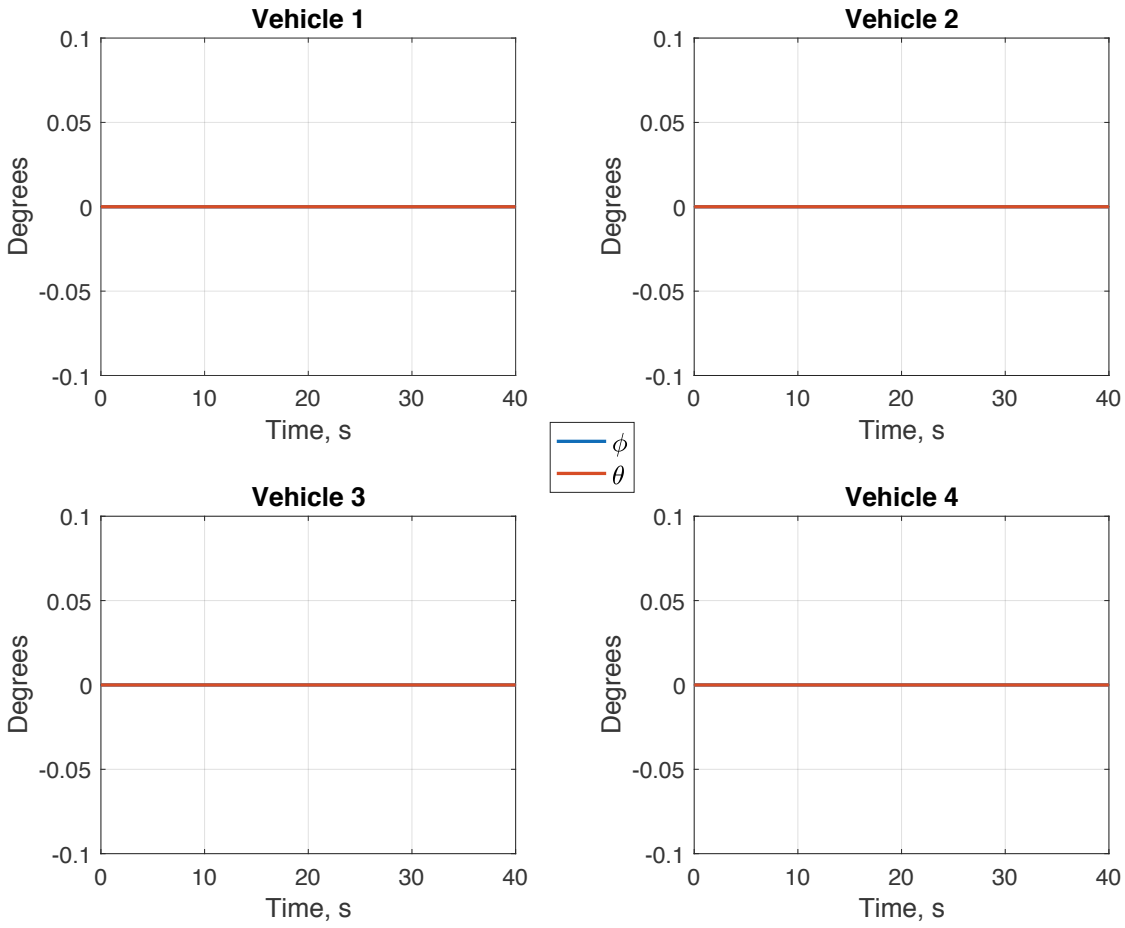

Figure 26 – Gimbal Histories for Active Kalman Filter: Symmetric Attach Points.

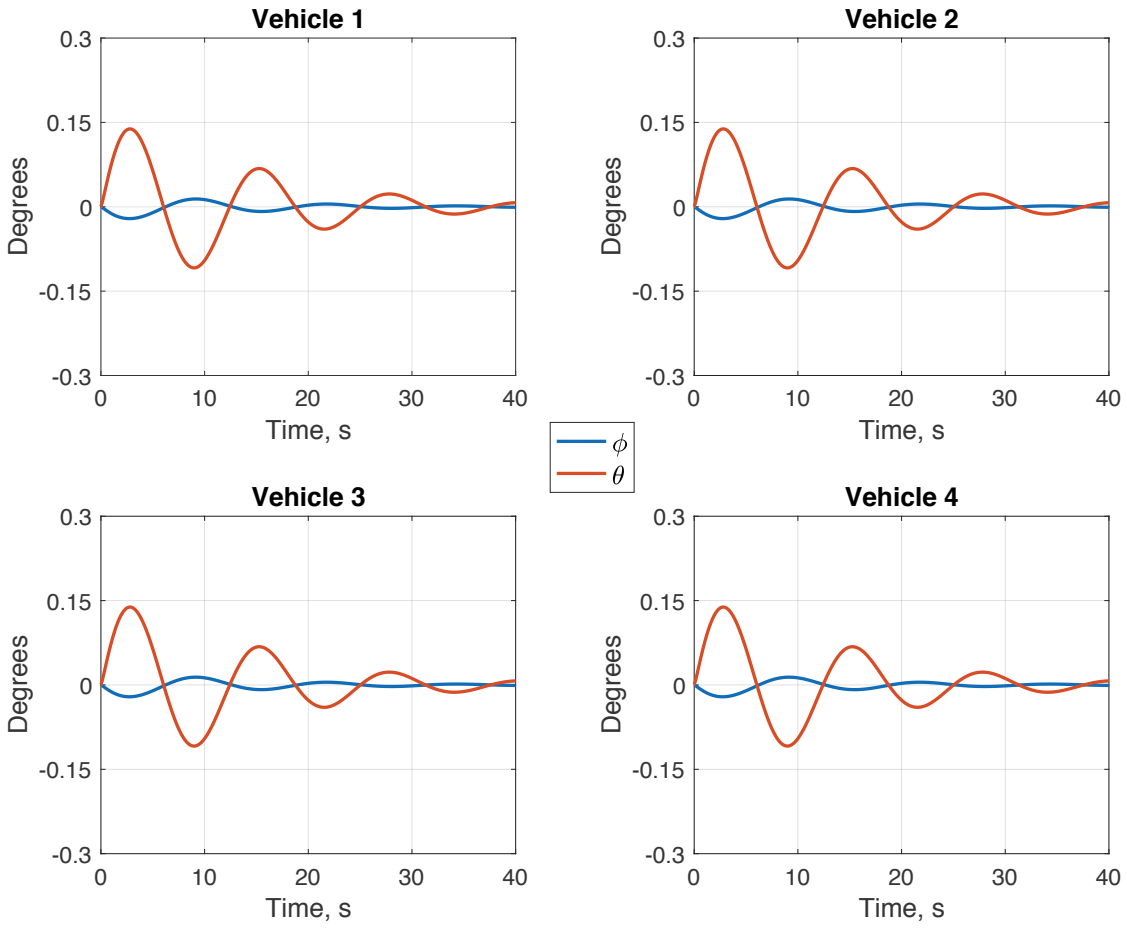

Figure 27 – Gimbal Histories for Active Kalman Filter: Asymmetric Attach Points.

The Kalman filter can only accurately update the augmented state when there is dynamic excitation in those states. As seen in Figure 22 and Figure 23, when the vehicles are attached in a symmetric pattern, the controller can easily and quickly achieve and maintain stability, severely limiting excitation to the payload states. Given near zero excitation error information, the Kalman filter does not change the radius vectors in this case from the nominal values assigned at time zero, as shown in Figure 28.

Opposing this, there is a short amount of time in the case with asymmetric attach points that the controller does not have immediate stability over the system. During this short period of time, the Kalman filter is able to start updating the radius vectors. However, if the Kalman filter has not achieved an error of zero for the radius vector estimates by the time the controller has stabilized the system within a small bound, updating slows and eventually stops with non-zero error, as shown in Figure 29. As soon as a trajectory is commanded, further excitation would allow the Kalman filter to further reduce these errors.

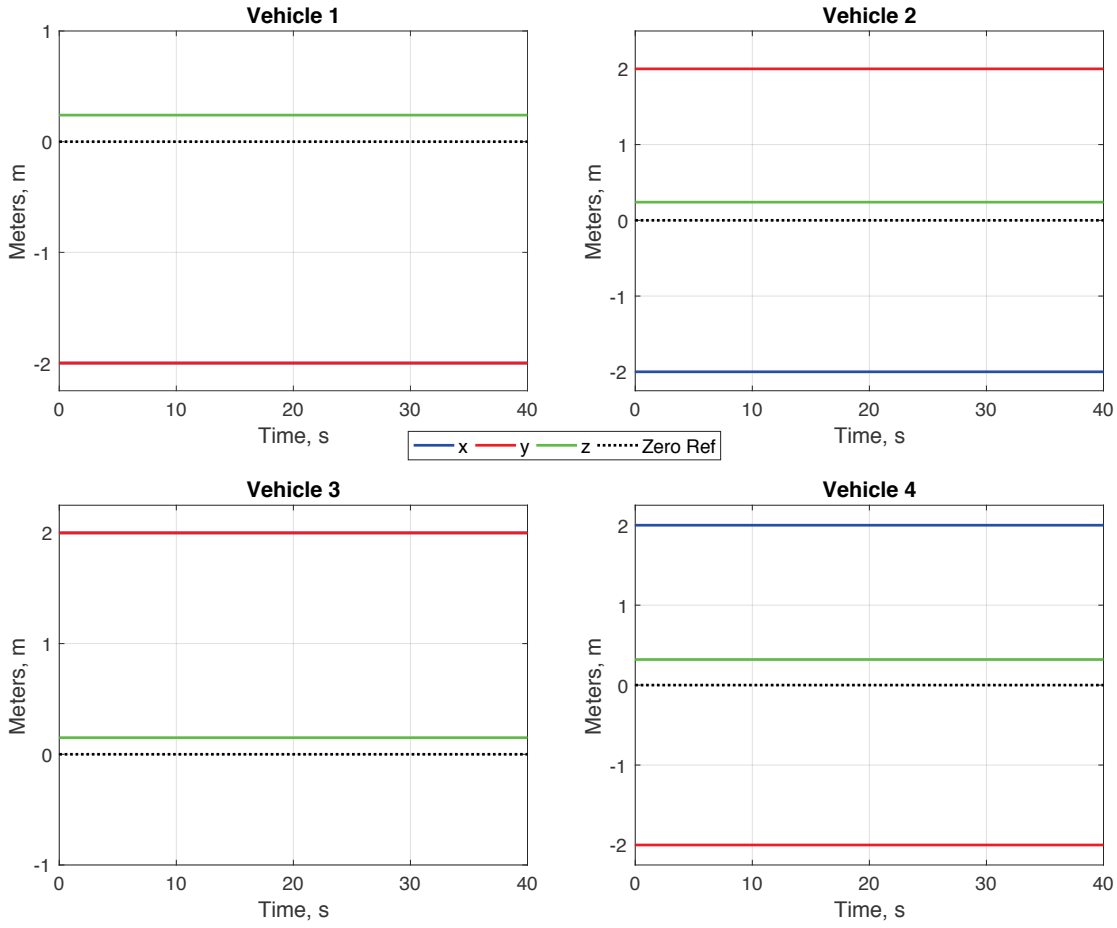

Figure 28 – Radius Vector Error Histories for Active Kalman Filter: Symmetric Attach Points.

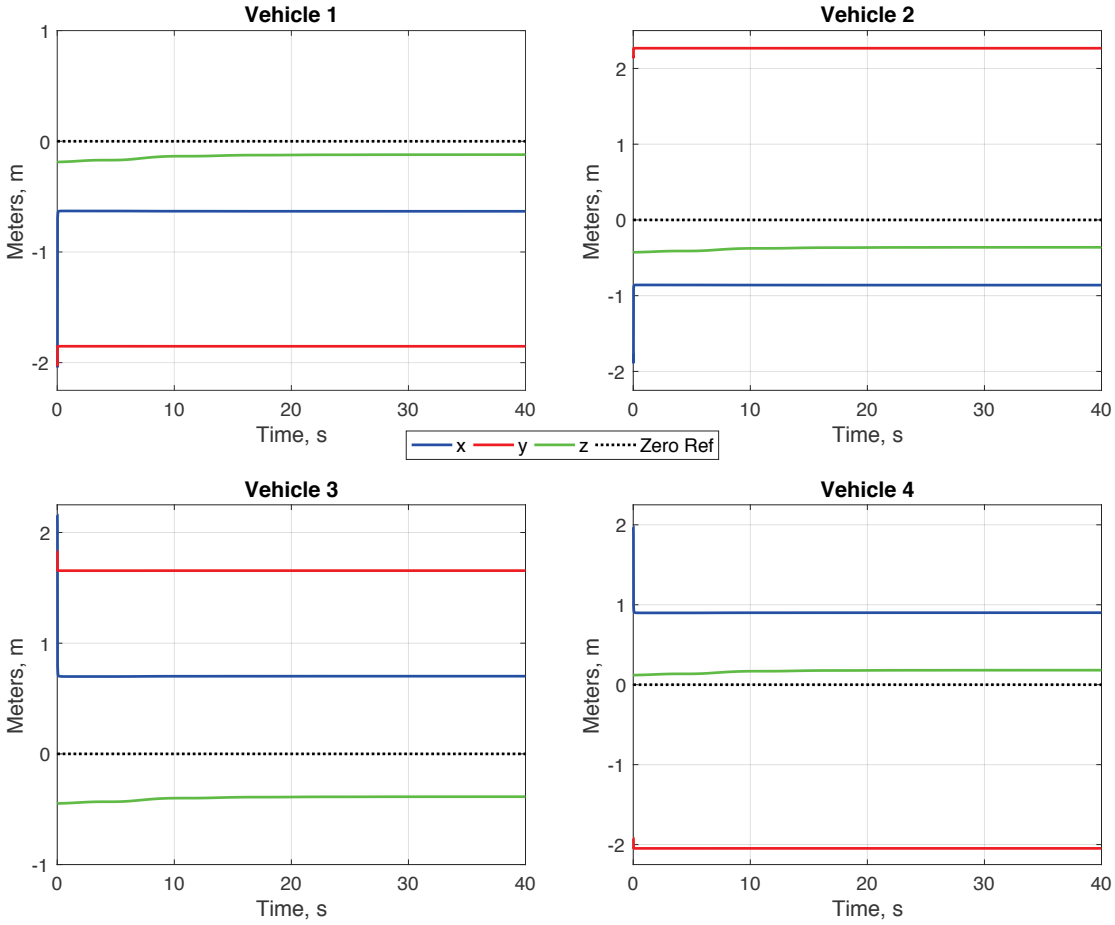

Figure 29 – Radius Vector Error Histories for Active Kalman Filter: Asymmetric Attach Points.

The radius estimates in the preceding cases converge very quickly towards stable values that satisfy the hover command. The tuning applied to the extended Kalman filter was tailored for this fast convergence. Figure 30 shows the radius error convergence history for a sample case that is tuned for greater uncertainty. Compared to the above case, the uncertainty in the initial guess for the radius vectors, related by the initial value given for the state error covariance matrix, was raised by eight orders of magnitude. Likewise, the measurement noise covariance matrix and the process noise covariance matrix were raised by ten and five orders of magnitude, respectively. This demonstrates a larger noise signal is expected from the sensors, as well as a greater uncertainty in the model dynamics being used for the filter.

The effect of raising these uncertainties is a slower convergence toward a steady state value. The radius convergence shown demonstrates that even though a specific use case may have a much wider uncertainty envelope, the proposed adaptive controller can still update values for a successful stabilization, even if the gains are tuned more poorly.

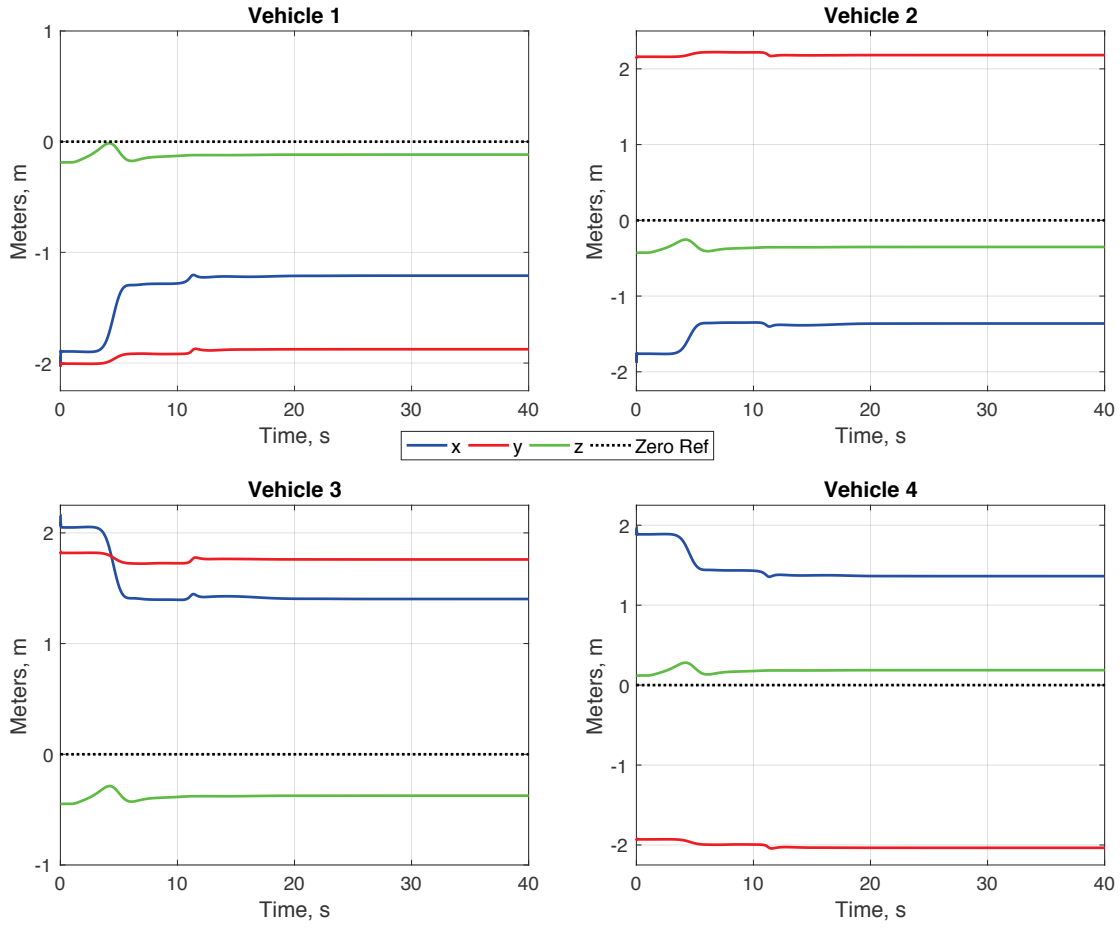

Figure 30 – Radius Vector Error Histories for Active Kalman Filter: Asymmetric Attach Points & Non-optimal tuning.

CHAPTER 5. TRADE STUDIES

A first trade study is provided comparing energy consumption of the set of modular vehicles in forward flight when carrying a bluff body payload versus an airfoil-shaped payload. Energy consumption metrics are quantified as a function of angle of attack and forward speed. Another trade study compares energy required to carry payloads of varying mass. In all cases presented here, when in forward flight the controller seeks to regulate the sideslip of the payload to zero. These studies show that energy consumption is reduced when using a payload that is airfoil shaped and operating at favorable angles of attack, in order to take advantage of the wing aerodynamics to produce lift.

Additional trade studies were performed to assess the capability of the system to stabilize a payload with randomized parameters. In these cases, Monte Carlo simulations are performed with weight, inertias, CG, and vehicle locations randomized to different degrees. The main goal of these cases is for the controller to stabilize the composite aircraft with no prior knowledge as to the mass, inertias, or configuration of the attached vehicles. This is to mimic a real-world scenario where the user has no precise a priori knowledge of payload or attachment characteristics.

5.1 Energy Consumption with Wing-Shaped Payload

5.1.1 Energy Consumption at Fixed Mass

A trade study was performed to compare energy expenditure while carrying a payload with a fixed mass of 70 kg. The comparisons were made for steady-state velocities varying from 9-21 m/s. In all simulations, a forward flight was commanded and energy

expenditure was recorded over a 60 second time window once the payload reached the commanded velocity within a tolerance of 0.05 m/s. First, a bluff body payload was simulated with a commanded pitch angle of zero. The same flight profile was then flown using the NACA0012 airfoil payload. Airfoil payload cases were additionally performed for varying angle of attack setpoints and commanded flight speeds.

Figure 31 shows the energy expenditure results from this trade study. As expected, the bluff body payload requires more energy than the airfoil case at nearly any speed, except for the zero angle of attack case at low speed. Replacing the bluff body with an airfoil-shaped container, a slight performance benefit is noted over the bluff body at zero angle of attack due solely to the reduction in payload drag. As the commanded angle of attack increases toward the maximum C_L/C_D, energy savings increase, with a savings of about 70% over the bluff body case when operated at 21 m/s and 14 degree angle of attack.

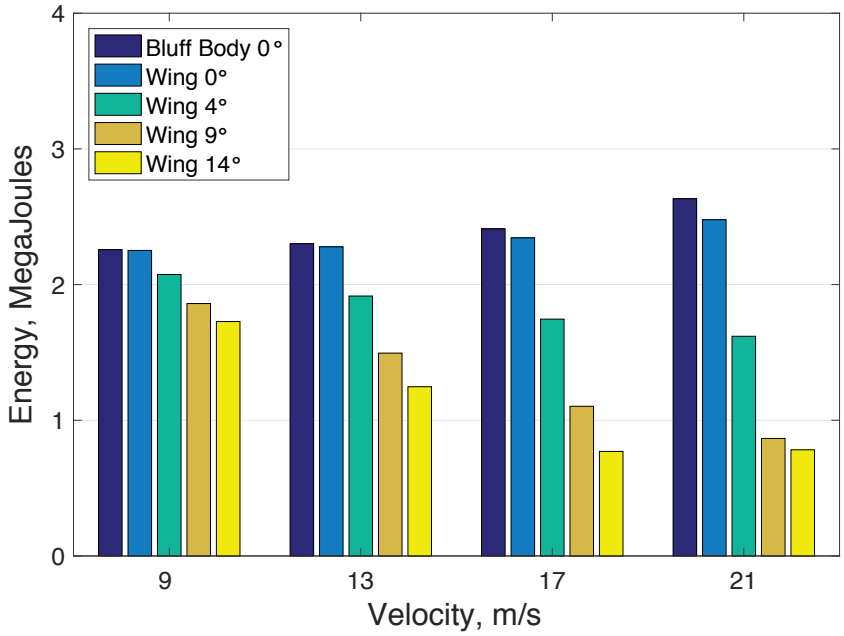

Figure 31 – Energy Expended for Various Velocities and Payload Configurations with a 70kg Payload. Increased lift from the payload results in reduced energy consumption as less thrust is needed to maintain altitude.

An additional benefit apparent from these results is that less energy is necessary to carry the payload at a faster speed. Delivering a payload sooner may be just as advantageous as expending less energy to complete the delivery. Thus, more deliveries can be completed in the same amount of time over greater distances than can be completed with a box-shaped or bluff-body payload.

5.1.2 Energy Consumption at Fixed Velocity

The second scenario involves transporting a payload at a fixed velocity of 22 m/s. The energy expenditure was again measured over a period of 60 seconds once the payload had reached the desired velocity. The mass of the payload was varied in increments of 10kg, starting at 60kg up to 100kg. A bluff body in level flight was simulated at each

payload weight to use as a basis for comparison. The airfoil-shaped container was then used to transport the payload, first with a fixed angle of attack of 10 degrees for each weight. Finally, the same simulations were performed with the angle of attack controller activated to optimize the lift generated for the given payload weight.

Figure 32 shows the results of this trade study. Note the large difference in energy necessary to carry the bluff body container versus the airfoil container. However, Figure 33 shows the more interesting result that the energy spent is relatively flat within the band of payload masses shown here. This effect is even more noticeable when the controller is allowed to vary the payload angle of attack. When the wing produces sufficient lift to counteract the vehicle weight, the thrust is responsible only for counteracting the composite vehicle drag. Since the drag coefficient of the airfoil payload changes relatively little in the band between 0 to 14 degrees angle of attack, the portion of horizontal thrust commanded will stay approximately the same across all payload masses when the velocity is held the same, as it only has to counteract the drag to maintain the desired velocity. The vertical component of the commanded thrust will always be close to zero if the angle of attack is controlled such that the lift of the payload compensates exactly for its weight.

The implications of this result are quite significant. For example, once an airfoil payload of a certain volume is constructed, there is essentially no benefit (from an energy standpoint) to operating it below a maximum weight equal to the total thrust of the set of modular vehicles. This maximum weight limit is obvoiusly necessary due to hover requirements. However, because the energy expenditure curve of the vehicles in forward flight is essentially flat, the payload can be filled to the maximum weight capacity with essentially the same expected energy expenditure.

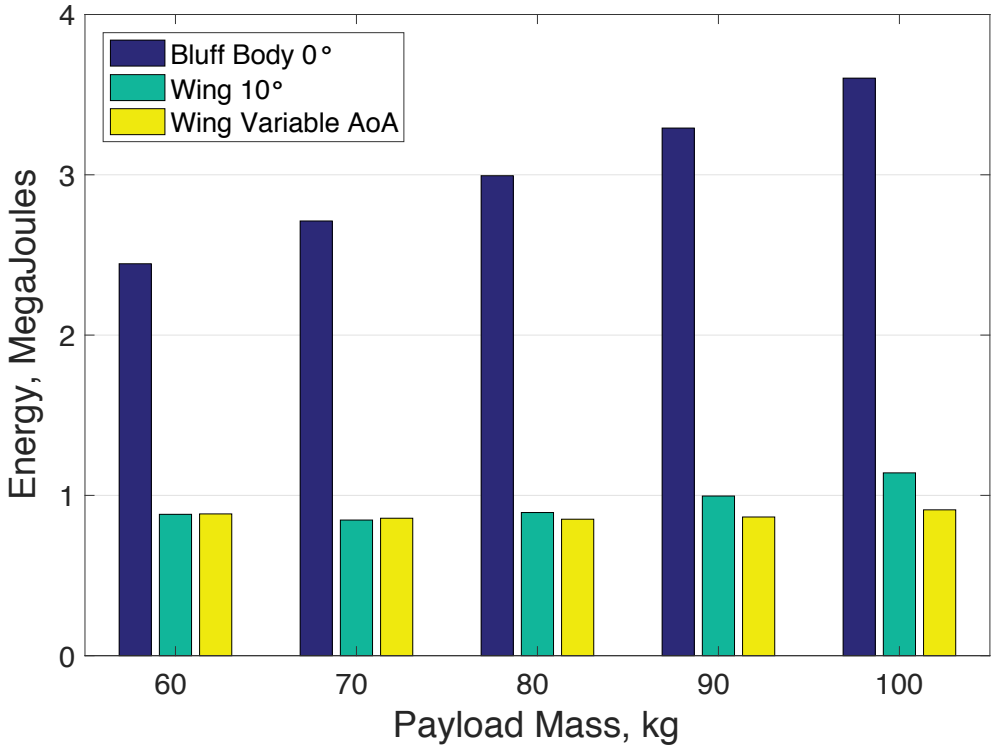

Figure 32 – Energy Consumption for Varying Payload Masses at Fixed Velocity. Total energy consumed for each payload mass.

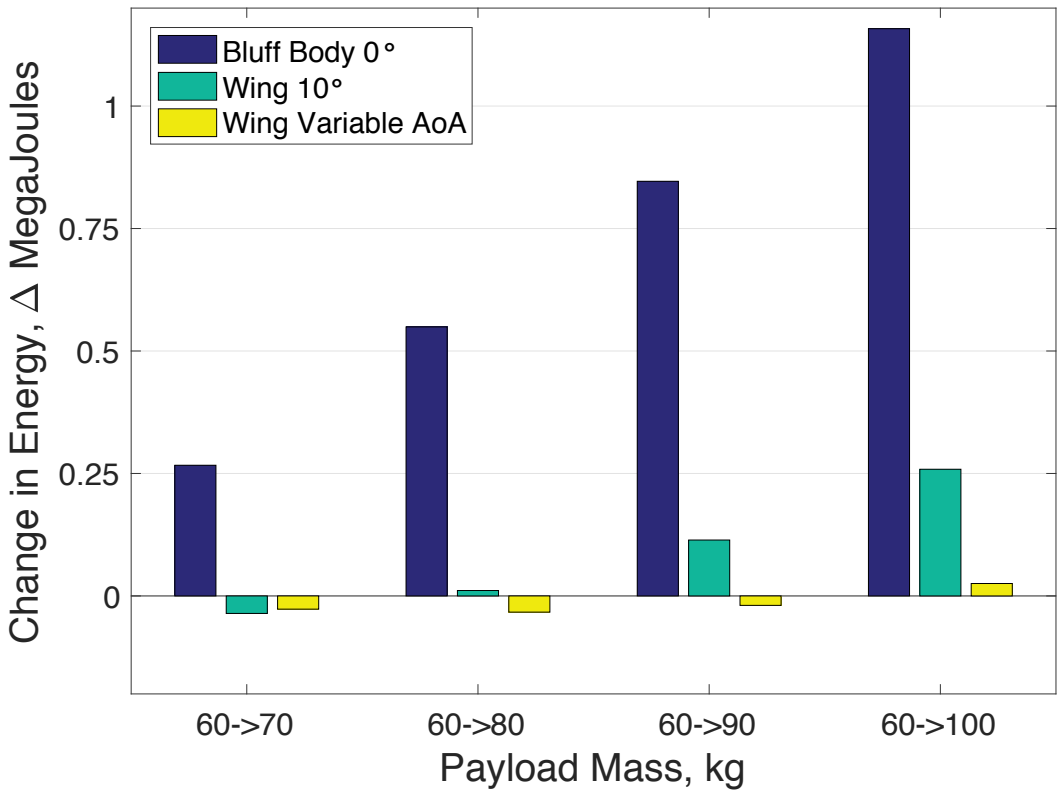

Figure 33 – Energy Consumption for Varying Payload Masses at Fixed Velocity. Change in energy consumed for varying payload masses, using the energy consumed during a 60 kg payload mission as the basis for comparison.

5.2 Monte Carlo Simulation of Stabilization Cases

As described above in the example trajectories subsection, there are three different cases useful in assessing the effectiveness of the Kalman filter. One case is where full knowledge of all system parameters is given to the controller, a second where no knowledge of true values is given, and finally a case with initial values subject to error, but the radius parameters are updated in real-time by the Kalman filter. A set of Monte Carlo simulations was run for each of these cases. Differing effects are achieved and demonstrated by varying the type of error perturbations given to the system.

Analysis of the filter effectiveness is done by quantifying the percentage of cases, out of 1000, that were able to successfully stabilize. Further, the time to stabilization and the altitude lost or gained until stabilization was achieved were also quantified. For all cases, the system was commanded to hover and the controller was to achieve linear and angular velocities below certain thresholds (near zero), and maintain those thresholds for eight seconds before reporting a successful stabilization.

In the cases below, three different methodologies may be used to randomly vary the radius parameters. The first method involves using the same mean error value, but randomizing the locations by some varying standard deviation with respect to this mean. The second method does the opposite, randomizing the mean error values while holding constant the standard deviation offset from these mean values. The third method randomly varies both the offset and the mean errors from zero, combining the other two methods. When using the first two methods, the rotor thrusts are not limited (saturated). When using the third method, rotor thrust saturation limits are imposed.

5.2.1 Error Perturbation

Figure 34-Figure 36 show a top down view of randomized radius vector endpoints on the payload for various types of error multipliers. A set of radius vectors was first created by varying only the standard deviation error from a given mean. Next, only the mean error was varied holding the standard deviation error constant. This was followed by a combination of both types of errors. By completing the scenarios with strictly standard deviation versus mean, it allowed assessment as to which type of perturbation had different effects on the percentage of cases that successfully stabilized. Shown in following subsections, having radius vectors that are farther away from the CG aids significantly in stabilization performance. This is in contrast to the reduction in stabilization performance due to increasing the standard deviation of the error. By combining the two types of perturbations trends, the results show which type of random perturbation has a larger influence.

In order to control the increment for enlarging either the standard deviation or mean error of the radius vectors, a separate error bound multiplier is used for each of these randomizations. To create the randomized points, first an initial normal distribution is created for each of the vehicles attach points. Then, to ensure one vehicle was assigned to each of the payload frame quadrants, these normal distributions are scaled to have a standard deviation of 0.2, such that all data points are between negative one and one. With the result, this distribution can alter the radius vector error from a nominal value, $r_{x,y,nom}$ used by the Kalman filter, to create a normal distribution of attach points strictly in the quadrant desired for each vehicle. Equations 55 and 56 describe how the attach point

distribution, $\mathrm{r}_{x,y,\oplus P\to Vi,Dist}$, is defined using a standard deviation multiplier, σ^2_{mult}, and mean multiplier, μ_{mult}, respectively. The distribution for the $\vec{I}_P$ direction value is shown, and the value for the $\vec{J}_P$ direction is similarly computed. Randomization in the $\vec{K}_P$ direction was restricted to within one meter above or below the payload CG, with no dynamic multiplier across cases. This was done so the effects of vehicle placement in the payload $\vec{I}_P - \vec{J}_P$ plane could be assessed.

$$\mathrm{r}_{x,\oplus P\to Vi,Dist.,std} = \sigma^2_{mult}\,\mathcal{N}(0, 0.2^2) + r_{x,nom} \tag{55}$$

$$r_{x,\oplus P\to Vi,Dist.,mean} = \mathcal{N}(0, 0.2^2) + \mu_{mult}\, r_{x,nom} \tag{56}$$

For the mixed multiplier, a distribution centered at a mean of one was used, such that one mixed error multiplier, η_{mult}, affects the standard deviation and mean simultaneously by,

$$\mathrm{r}_{x,\oplus P\to Vi,Dist.,mix} = \eta_{mult}\left(\mathcal{N}(0, 0.2^2) + r_{x,nom}\right) \tag{57}$$

Figure 37-Figure 39 show the final statistical values of the set of radius vector errors used for each of these scenarios. These error statistics vary slightly as they are computed using a normal distribution with a low standard deviation applied to the payload CG as well.

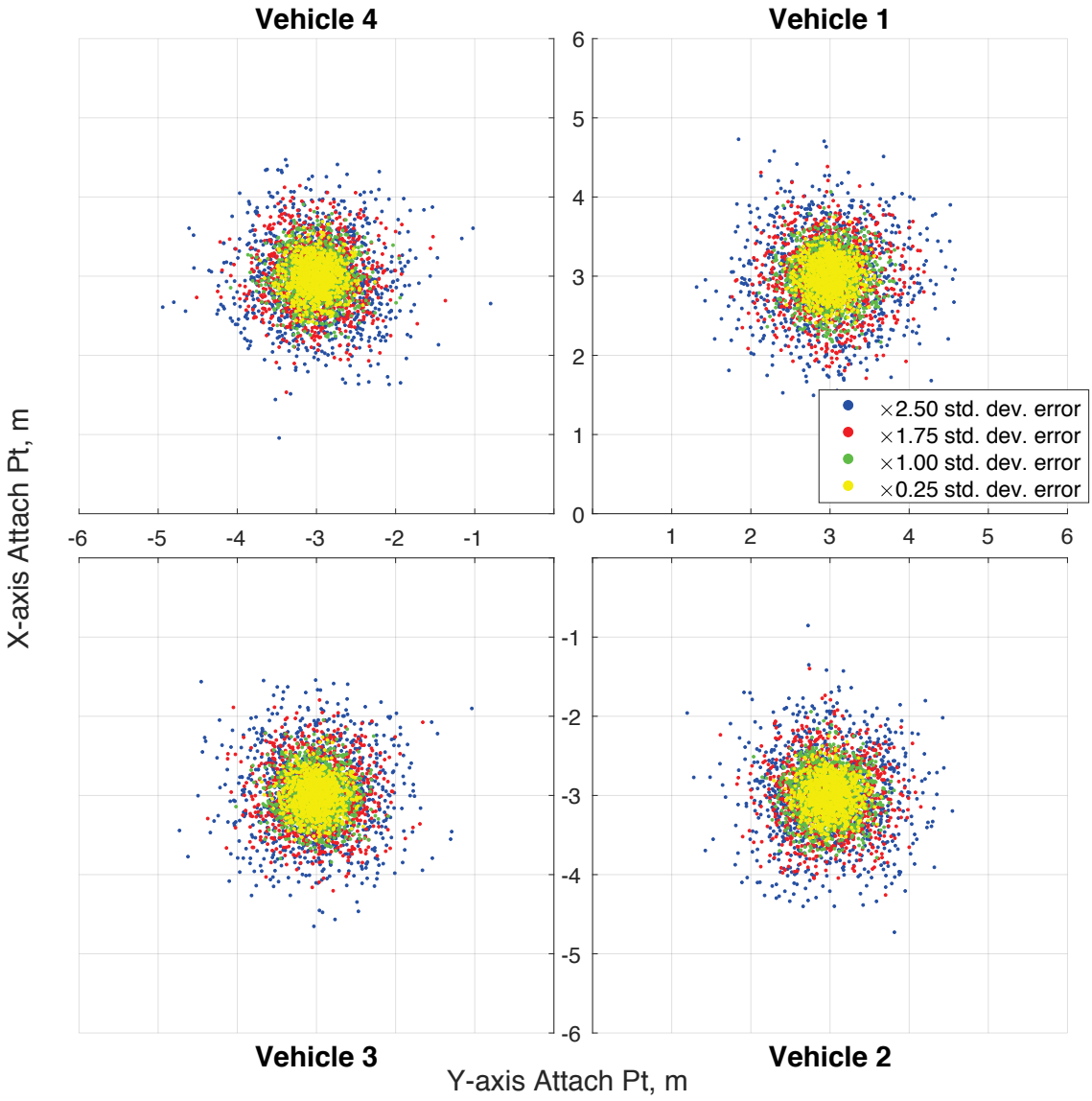

Figure 34 – Types of Radius Vector Errors: Standard Deviation Error. Standard deviation of error is randomized, with constant mean error.

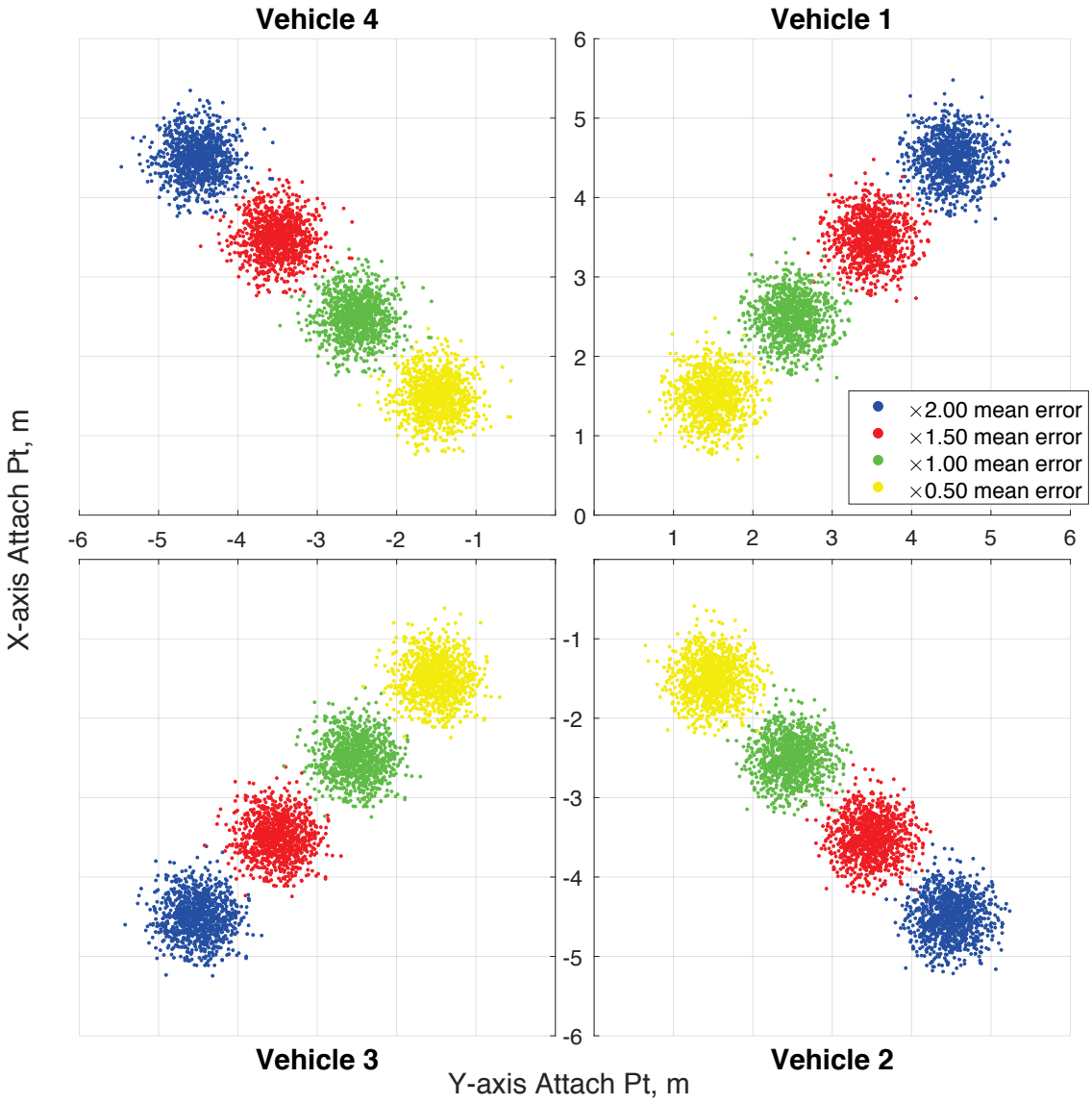

Figure 35 – Types of Radius Vector Errors: Mean Error. Mean error is randomized, with constant standard deviation.

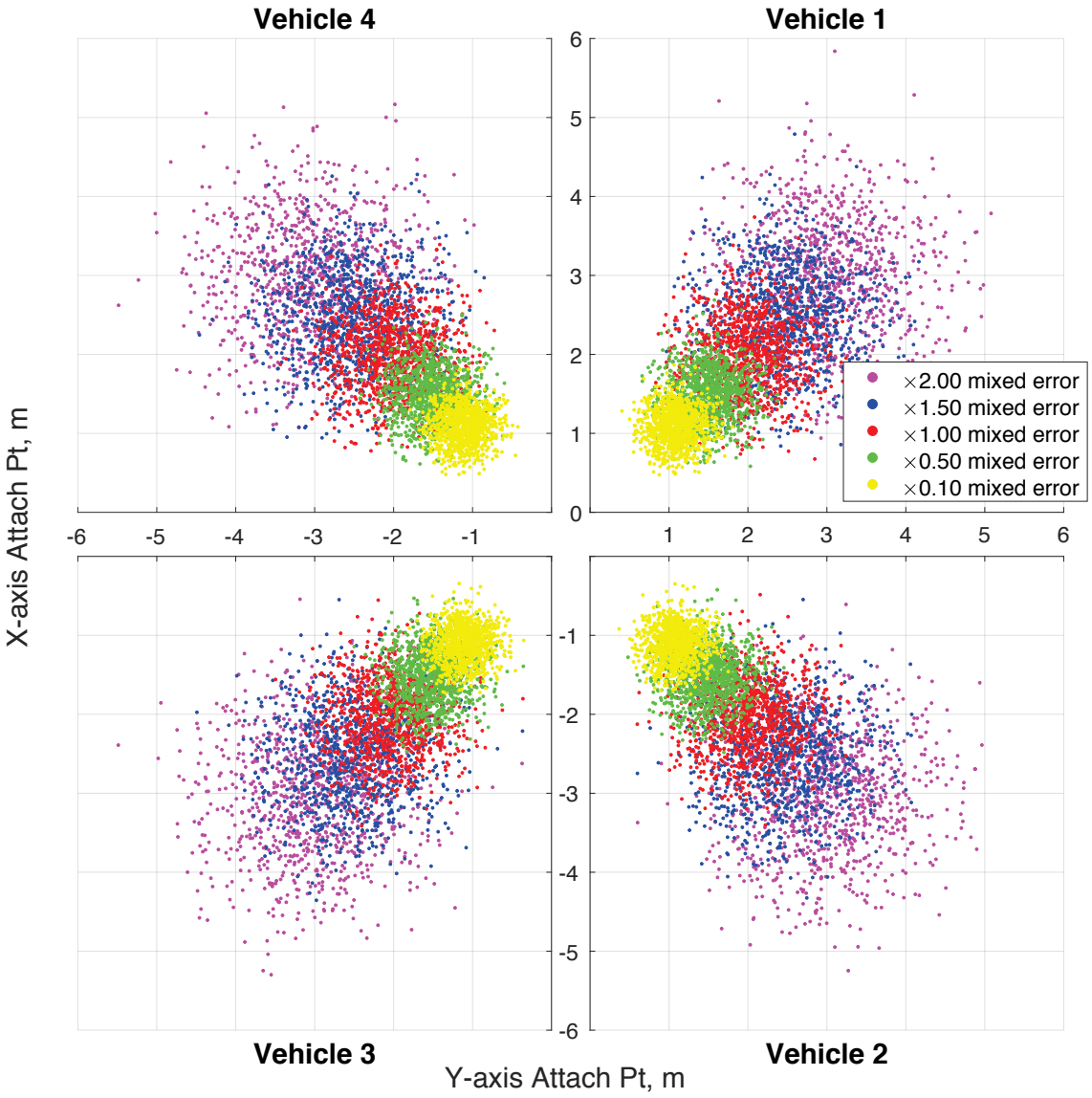

Figure 36 – Types of Radius Vector Errors: Mixed Error. Both standard deviation and mean of error are randomized.

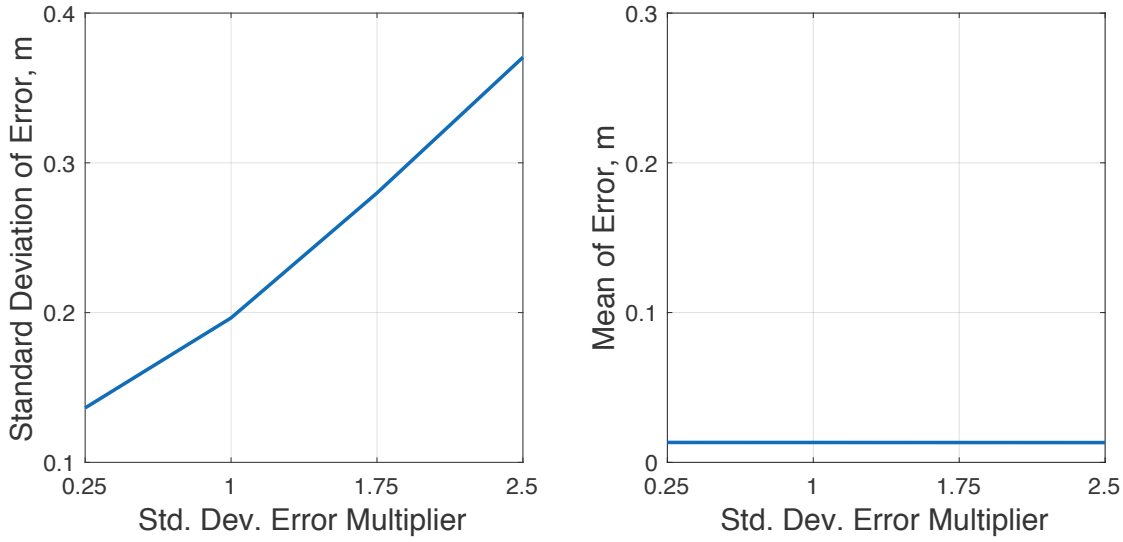

Figure 37 – Statistics of Radii Errors: Standard Deviation Error. Standard deviation of error is scaled, with constant mean error.

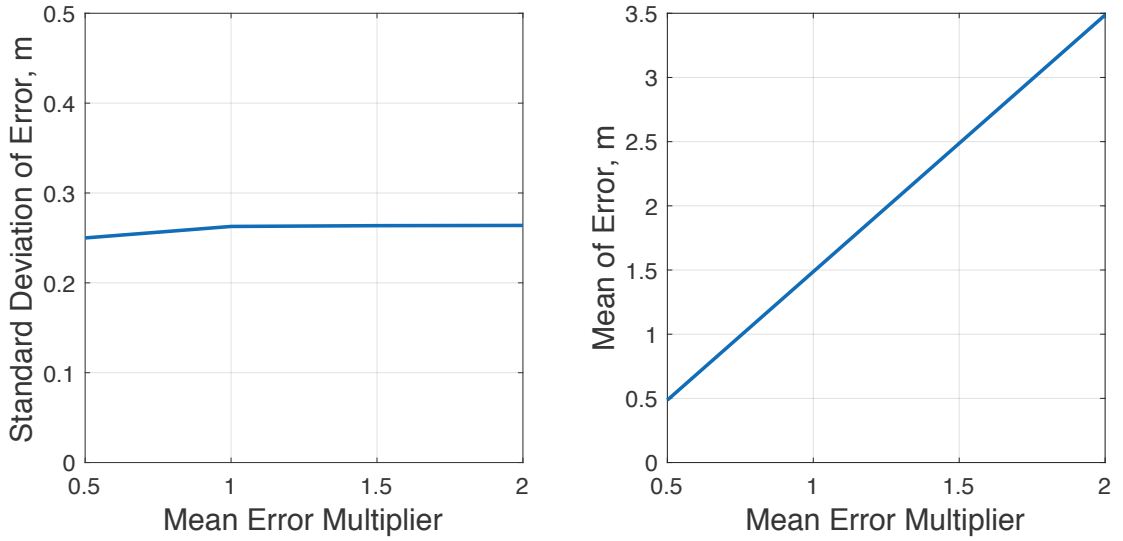

Figure 38 – Statistics of Radii Errors: Mean Error. Mean error is scaled, while maintaining constant standard deviation.

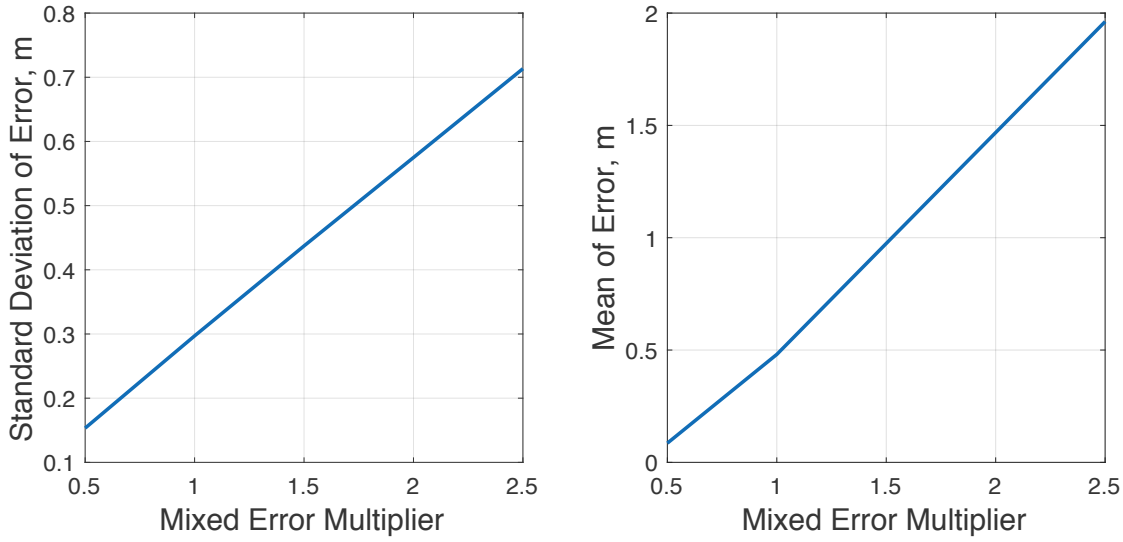

Figure 39 – Statistics of Radii Errors: Mixed Error. Mean and standard deviation error are scaled simultaneously.

5.2.2 Percentage of Cases that Successfully Stabilized

Figure 40 shows the percentage of cases that successfully stabilized with saturation limits active and both mean and standard deviation errors added to the radius vector perturbations. The full knowledge cases were generally successful, ranging between 84-88% success, as were the Kalman filter cases. The cases with no knowledge achieved stabilization with 21-28% success.

At a mixed error multiplier, η_{mult}, equal to 1.5, the Kalman filter cases have visibly more success than the perfect knowledge cases. This is due to the ability of the Kalman filter to compensate for rotor saturation by adjusting the control allocation matrix. As shown in Figure 41 and Figure 42, the Kalman filter scenario has a large number of cases that, at some point during the stabilization process, command the motors above their

saturation limits. Because the Kalman filter adjusts the radius vectors used in the controller if a measured output does not match the expected output, it can avoid saturating the motors for long periods of time. Figure 42 relates this by showing the percentage of all motor commands that were above the saturation limit, for each of the three cases. On the other hand, for the few cases where the full knowledge cases do saturate the motor commands, the controller continues to command those saturated values because the control allocation scheme uses fixed parameters. This effectively reduces its success envelope since it is not as flexible.

Finally, the non-updating scenario is more successful with larger mean error multipliers. As shown in Figure 43 and Figure 44, the case with larger mean error, having larger moment arms available to multiply by the thrust values, raises the stabilization performance of the system. Larger standard deviation errors reduce stabilization performance, but not to the same extent as the amount gained for having larger moment arms in the mean error case. This imbalance is the reason that more cases yield stable flight when the mixed error multipliers are larger in the fixed-parameter, no-knowledge case. Because of the larger moment arms, the total number of successful cases that had some saturation was lower, as the controller did not need larger thrust values to achieve the same moments on the body necessary for stabilization.

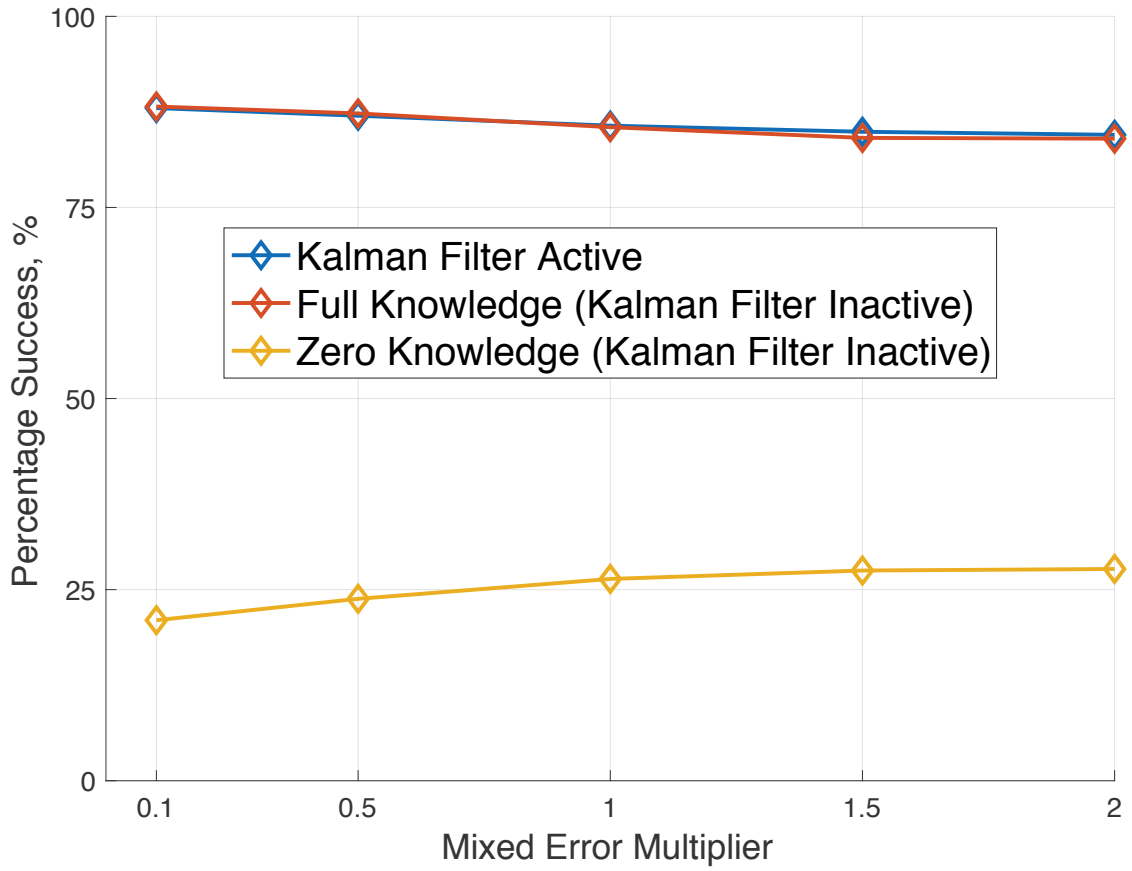

Figure 40 – Percentage of Cases that Successfully Stabilized: Mixed Error.

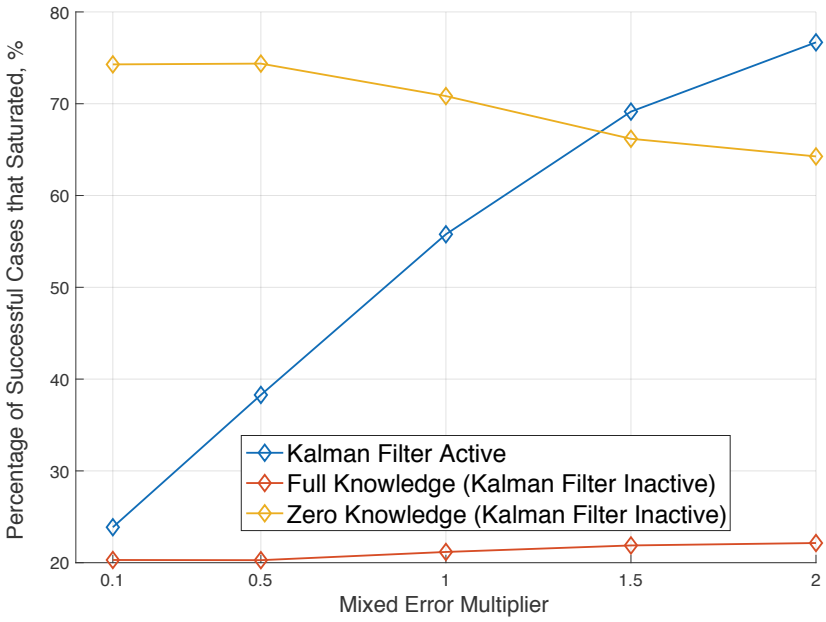

Figure 41 – Percentage of Successful Cases with Saturated Motor Commands.

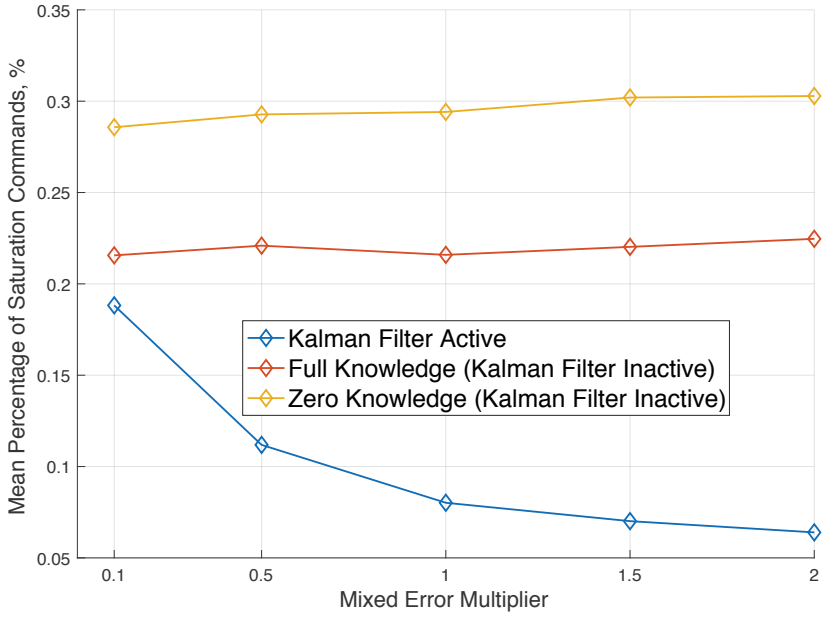

Figure 42 – Mean Percentage of All Motor Commands that Saturated.

Figure 43 – Percentage of Cases that Successfully Stabilized: Standard Deviation Error.

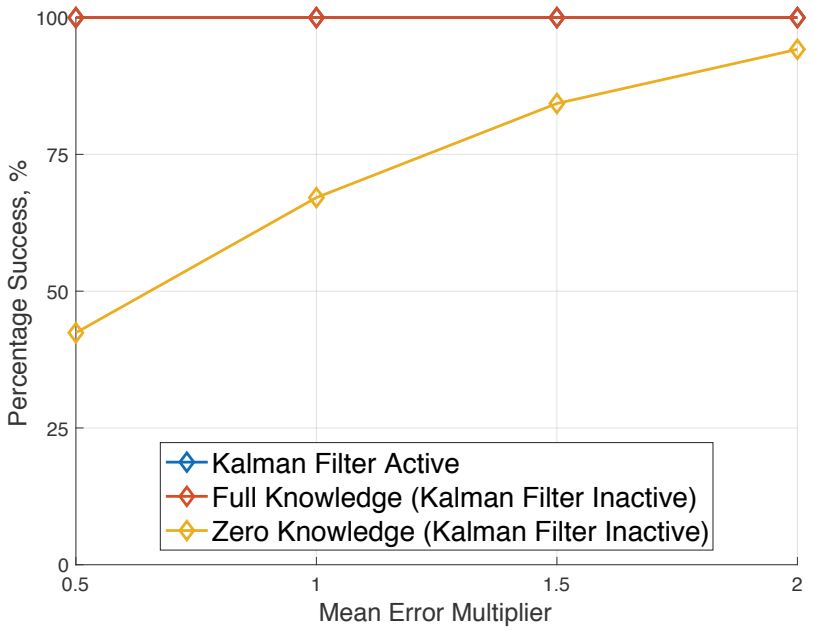

Figure 44 – Percentage of Cases that Successfully Stabilized: Mean Error.

5.2.3 Average Time to Stabilize and Altitude Error

Figure 45 and Figure 46 are related to each other in that the longer it takes for the system to stabilize, the farther it will fall in altitude before recovering hover. Despite the necessity of updating radius vectors, the Kalman filter case performs similarly to the full knowledge case in terms of the ability to stabilize quickly. The case where no knowledge is passed to the controller decreases time to stabilize at higher mixed error multipliers.

When the weight bias given to the controller is above the randomized true value, the altitude error will trend positive until the PID can compensate for the difference in expected mass. Likewise, the altitude error will trend negative if the bias is lower than the true system value for any case. The absolute value of these altitude errors was used to compute the mean altitude error so as to avoid the positive and negative values cancelling when reporting the aggregate value.

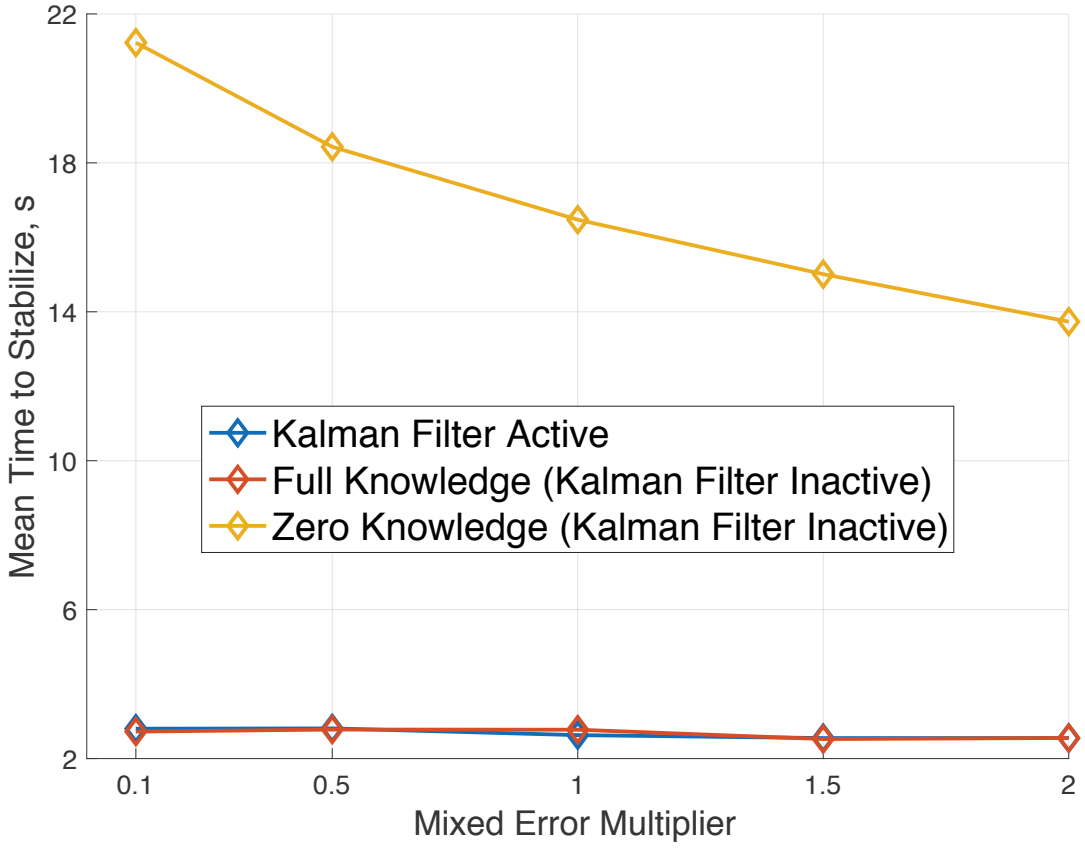

Figure 45 – Average Time to Stabilize with Mixed Error.

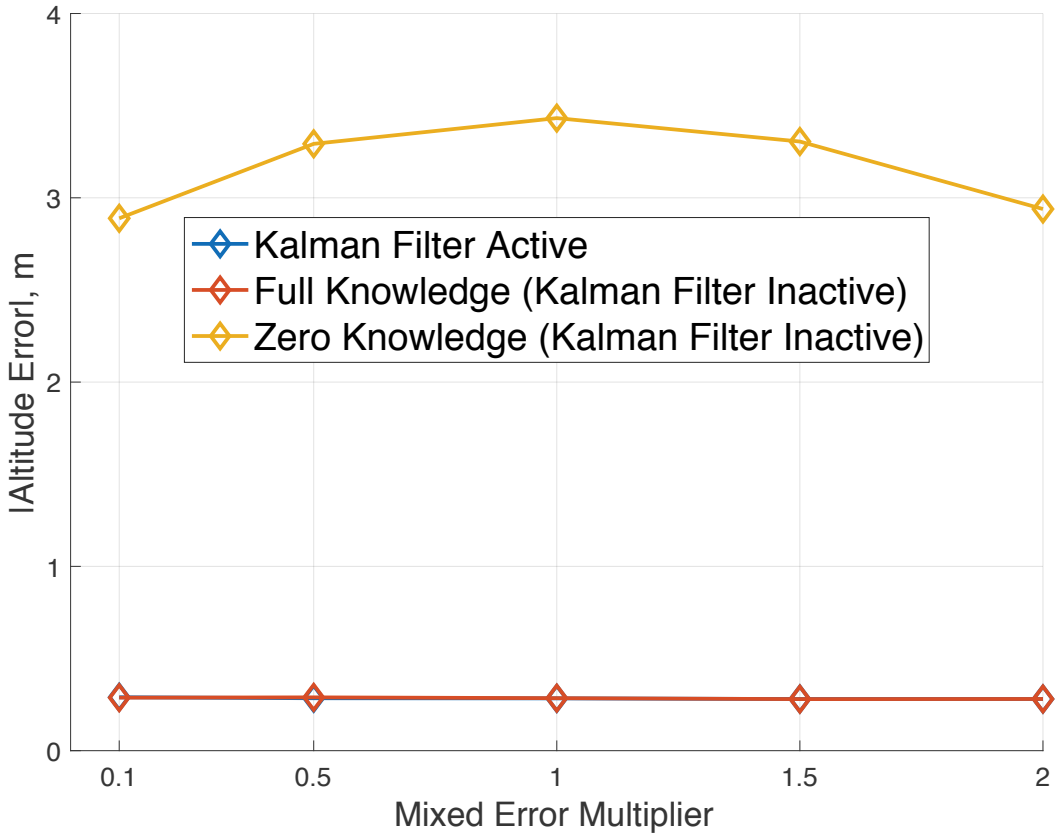

Figure 46 – Altitude Error Incurred During Stabilization with Mixed Error.

CHAPTER 6. PROTOTYPE DESIGN AND CONSTRUCTION

To establish feasibility and gather experimental data for the simulation model, a prototype heavy-lift multirotor vehicle prototype has been designed and constructed. The design uses an X-configuration, where each arm is approximately 0.53 meters from rotor hub to rotor hub. Two coaxial rotors are located at the end of each arm, leading to a total of 8 rotors. Note that the primary reason this design is selected is that it allows for increased thrust capability (through coaxial rotors) without increasing the overall size of the vehicle. This is important as it allows the collective vehicle set to attach to smaller payloads, on the order of 2 meters or less. The motors selected for this prototype are ElectriFly Rimfire 1.20 outrunner motors with a KV rating of 450 RPM/V attached to propellers of 0.406 m diameter. At higher throttle ranges, experimental tests showed that one motor mounted in a coaxial configuration draws about 80 A, and thus KDE Direct KDEXF-UAS95HVC electronic speed controllers were selected with a continuous current rating of 95 A and burst capability up to 170 A. Four lithium-polymer batteries of 5500 milli-Amp hour capacity each are mounted to the center of the vehicle, with each battery responsible for powering one pair of coaxial motors. A photo of this vehicle is shown in Figure 47.

Figure 47 – Prototype X-8 Octocopter.

Several benchtop thrust experiments were performed with the above electronics configuration to characterize thrust performance and predict overall lifting capacity of the multirotor vehicle. First, total thrust vs throttle percentage was measured for a single rotor, as shown in Figure 48. Similar experiments were performed with the coaxial rotor combination at a rotor spacing distance of 0.292 m, with the results shown in the same figure. Comparing these curves, it is clear that adding the coaxial rotor resulted in thrust increases of about 50-70% over the single-rotor case, depending on the throttle percentage. A second experiment was performed to quantify the benefit of spinning each rotor in the coaxial pair in opposite directions. Figure 49 shows thrust produced vs throttle percentage. The comparison is made by using the ratio of thrust from a single rotor to coaxially mounted co-rotating rotors, versus the ratio of a single rotor to coaxially mounted counter-rotating rotors. Co-rotating involves coaxial rotors spinning in the same direction while

counter-rotating involves two coaxial rotors spinning in opposite directions. When spinning in the same direction, the coaxial rotor pair provides about 1.4 times the thrust available with a single rotor, whereas when the rotors are counter-rotating, the coaxial configuration provides about 1.6 times. This increase in thrust using counter-rotating propellers is caused by the increase in blade angle of attack over the lower rotor due to swirl effects in the wake from the upper rotor. Given these experimental measurements, it was determined that the prototype vehicle will be configured with four coaxial, counter-rotating rotor pairs. Using this configuration, the experimental results shown in Figure 48 lead to an estimate of 380 N of total thrust for the prototype vehicle.

The vehicle was then assembled, and as can be seen in Figure 50, the total experimental thrust capability was consistently measured to be 400 N. The vehicle weight, including all four batteries, eight motors, eight electronic speed controllers, control board, and frame was measured as 107.6 N. This produces a total payload capacity of 292 N, or 29.7 kg. It is envisioned that vehicles of this general size and thrust capacity can be used to transport payloads of meaningful weight using similar control schemes to those described above.

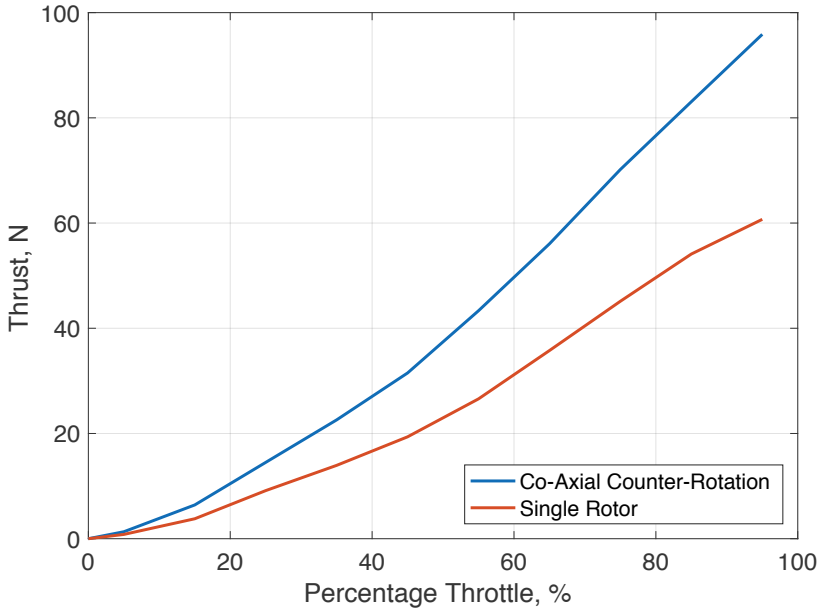

Figure 48 – Experimental Results for Thrust vs. Throttle Percentage: Single Rotor vs Co-Axial Rotors.

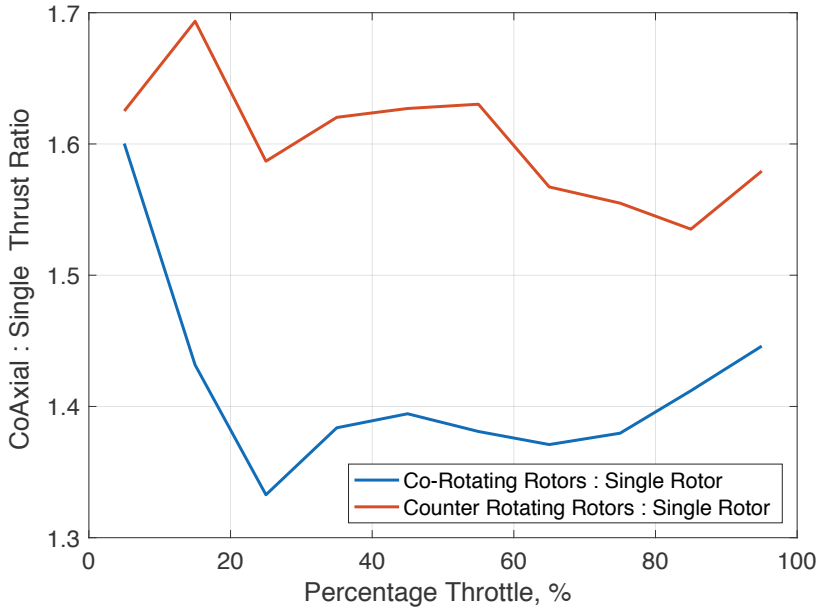

Figure 49 – Experimental Results for Thrust vs. Throttle Percentage: Co-rotating vs Counter-rotating Rotors.

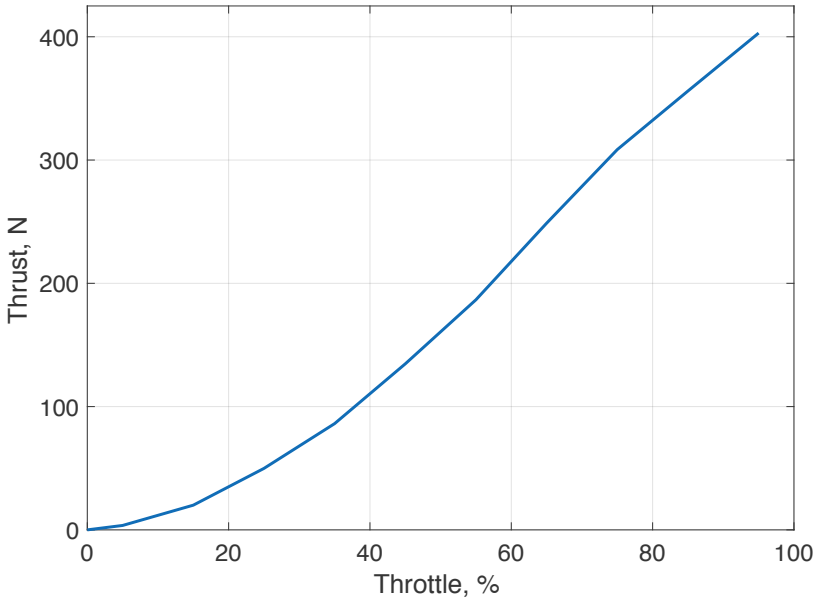

Figure 50 – Gross Experimental Thrust of X-8 Octocopter.

CHAPTER 7. CONCLUSION

A modeling and simulation framework for modular vertical lift air vehicles has been proposed. This new type of air vehicle configuration is comprised of multiple vertical lift vehicles which rigidly attach to a payload, forming a composite aircraft capable of vertical flight. In addition to presenting an overall simulation methodology, a flight control law is proposed for the composite aircraft that relies on control allocation to generate control commands for each attached vehicle. An augmented form of the controller allows the specification of a desired payload angle of attack in forward flight, which can be used to optimize energy consumption when using payloads with favorable aerodynamic characteristics such as wings.

Further, an extended Kalman filter can be used to widen the operating envelope, adapting the controller for situations where payload parameters are not known prior to flight. Simulation results show that the proposed control law is capable of stabilizing the composite vehicle throughout a payload transportation mission. Trade studies also show that a significant energy savings is possible when using payload containers which are specifically designed to produce lift in forward flight. Monte Carlo based trade studies also demonstrate that use of a Kalman filter can greatly improve stabilization performance. The improvement in probability of stabilization can be three times higher than for situations where no knowledge of system parameters is given to the payload. Overall, results show that the modular vertical lift system is feasible for payload delivery missions for certain payload weight classes and desired travel distances

APPENDIX A. PARAMETER VALUES

Table 1 – Parameter Values Used in Simulation.

δt	0.01 s
$\vec{C}_{d,vi}$	[2.5,2.5,3.5]
$m_{payload}$	70 kg
m_{Vi}	11 kg
m_{system}	114 kg
C_l	From NACA0012 table
C_d	From NACA0012 table
A	2.74 m^2
C_s	0.4
A_s	0.082 m^2
c_T	1.3841×10^{-4}
c_q	3.257×10^{-6}
ω_{max}	855 RAD/s
ω_{min}	110 RAD/s
$\left\|\vec{r}_{\oplus P \to \oplus Vi}\right\|$	1.2 m
$\left\|\vec{r}_{\oplus V \to dj}\right\|$	0.363 m
$Inertia_{payload}$	[38.3,18.1,55,8]
$Inertia_{vi}$	[0.28,0.28,0.57] $kg \cdot m^2$
err_{th}	80 m
v_{max}	23 m/s
α_{init}	6 degrees

REFERENCES

[1] Schmidt, M. D., "Simulation and Control of a Quadrotor Unmanned Aerial Vehicle," Master's Thesis, Paper 93, Electrical Engineering Dept., Univ. of Kentucky, Lexington, KY, 2011.

[2] Leishman, R. C., Macdonald, J., Beard, R. W., and McLain, T. W., "Quadrotors and Accelerometers: State Estimation with an Improved Dynamic Model," IEEE Control Systems, Vol. 34, Issue 1, 2014, pp. 28-41.

[3] Mahony, R., Kumar, V., Corke, P., "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," IEEE Robotics & Automation Magazine, Vol. 19, Issue 3, 2012, pp. 20-32.

[4] M. Maisel, D. Giulianetti, D. Dugan, The History of the XV-15 Tilt Rotor Research Aircraft: From Concept to Flight, Monographs in Aerospace History #17, National Aeronautics and Space Administration, Washington, DC, 2000.

[5] Trouwborst C., "Control of Quadcopters for Collaborative Interaction," Report nr. 006RAM2014, University of Twente, Enschede, Netherlands, 2014.

[6] Wolin, J. A., "On Designing Collaborative Robotic Systems with Real-Time Operating Systems and Wireless Networks," Master's Thesis, Computer Engineering Dept., Boise State University, Boise, Idaho, 2014.

[7] Waslander, S. L., Hoffmann, G. M., Jank, J. S., and Tomlin, C. J., "Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, New York, 2005, pp. 3712-3717.

[8] Milionis, G., "A Framework for Collaborative Quadrotor – Ground Robot Missions," Master's Thesis, Physics Dept., Naval Postgraduate School, Monterey, CA, 2011

[9] Mellinger, D., Shomin, M., Michael, N., and Kumar, V., "Cooperative Grasping and Transport using multiple Quadrotors," *Distributed Autonomous Robotic Systems*, edited by A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, et al.,

Springer Tracts in Advanced Robotics, Vol. 83, Springer Berlin Heidelberg, Heidelberg, Germany, 2013, pp. 545-558.

[10] Wu, G., and Sreenath, K., "Geometric Control of Multiple Quadrotors Transporting a Rigid-body Load," *53rd IEEE Conference on Decision and Control*, IEEE, New York, 2014, pp. 6141-6148.

[11] Bangura, M., Mahony, R., "Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor," *Australasian Conference on Robotics and Automation*, Red Hook, NY, 2012, pp. 115-124.

[12] Qu, Y., Zhu, X., and Zhang, Y. M., "Cooperative Control for UAV Formation Flight Based on Decentralized Consensus Algorithm," *Intelligent Robotics and Applications: 5th International Conference, ICIRA, Part 1*, edited by C. Y. Su, S. Rakheja, and H. Liu, Springer Berlin Heidelberg, Heidelberg, 2012, pp. 357-366.

[13] 02Tjønnås, J., Johansen, T. A., "Adaptive control allocation," Automatica, Vol. 44, Issue 11, November 2008, pp. 2754-2765.

[14] 07Casavola, A., Garone, E., "Fault-tolerant adaptive control allocation schemes for overactuated systems," International Journal of Robust and Nonlinear Control, Wiley, Vol. 20, Issue 17, November 2010, pp. 1958-1980.

[15] 011Zhang, Y., Suresh, V. S., Jiang, B., Theilliol, D. "Reconfigurable Control Allocation against Aircraft Control Effector Failures," *16th IEEE International Conference on Control Applications*, IEEE, Singapore, 2007.

[16] 014Burken, J. J., Lu, P., Wu, Z., Bahm, C., "Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation," Journal of Guidance, Control, and Dynamics, AIAA, Vol. 24, No. 3, 2001.

[17] 01 Casavola, A., Garone, E., "Adaptive fault tolerant actuator allocation for overactuated plants," *Proceedings of the 2007 American Control Conference*, IEEE, New York, 2007.

[18] 01aPorter, R., Shirinzadeh, B., Choi, M., "Modelling and Daisy Chaining Control Allocation of a Multirotor Helicopter with a Single Tilting Rotor," Electronics, MDPI, Vol. 5, Issue 4, 2016.

[19] 01bKendoul, F., Fantoni, I., Lozano, R., "Modeling and control of a small autonomous aircraft having two tilting rotors," *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, IEEE, Spain, 2005.

[20] 01cRyll, M., Bülthoff, H. H., Giordano, P. R., "Modeling and Control of a Quadrotor UAV with Tilting Propellers," *2012 IEEE International Conference on Robotics and Automation*, IEEE, Minnesota, 2012.

[21] 01gJin, J., "Modified Pseudoinverse Redistribution Methods for Redundant Controls Allocation," Journal of Guidance, Control, and Dynamics, AIAA, Vol. 28, No. 5, 2005.

[22] 03Durham, W. C., "Constrained Control Allocation," Journal of Guidance, Control, and Dynamics, AIAA, Vol. 16, No. 4, 1993.

[23] Brown, A., Rogers, J., "A Sampling-Based Probabilistic Path Planner for Multirotor Air Vehicles in Cluttered Environments," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, March 2016, 0954410016636912.

[24] Munson, Bruce R., Okiishi, Theodore H., Huebsch, Wade W., and Rothmayer, Alric P., *Fundamentals of Fluid Mechanics*, 7th ed., John Wiley & Sons, Inc., Hoboken, NJ, 2013, pp. 529-530.

[25] Taylor, B., Rogers, J., "Experimental Investigation of Helicopter Weight and Mass Center Estimation," *51st AIAA Aerospace Sciences Meeting*, AIAA, January 2013.